



TUGAS AKHIR - KI141502

IMPLEMENTASI K-MEANS BERBASIS GRAPH UNTUK PENGELOMPOKAN PENELITI PADA SISTEM REPOSITORI RISET ITS

**HIFNIE BILFASH
NRP 51131000091**

Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.

Dosen Pembimbing II
Anny Yuniarti, S.Kom, M.Comp.Sc

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

IMPLEMENTASI K-MEANS BERBASIS GRAPH UNTUK PENGELOMPOKAN PENELITI PADA SISTEM REPOSITORI RISET ITS

**HIFNIE BILFASH
NRP 5113100091**

**Dosen Pembimbing I
Diana Purwitasari, S.Kom., M.Sc.**

**Dosen Pembimbing II
Anny Yuniarti, S.Kom, M.Comp.Sc**

**Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESES - KI141502

IMPLEMENTATION OF GRAPH BASED K-MEANS CLASSIFICATION FOR ITS RESEARCH REPOSITORY SYSTEMS

**HIFNIE BILFASH
NRP 5113100091**

**First Advisor
Diana Purwitasari, S.Kom., M.Sc.**

**Second Advisor
Anny Yuniarti, S.Kom, M.Comp.Sc**

**Department of Informatics
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

IMPLEMENTASI K-MEANS BERBASIS GRAPH UNTUK PENGELOMPOKAN PENELITI PADA SISTEM REPOSITORI RISET ITS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

HIFNIE BILFASH

NRP: 5113100091

Disetujui oleh Pembimbing Tugas Akhir

1. Diana Purwitasari, S.Kom., M.Sc.
(NIP. 19780410200312200) (Pembimbing 1)
2. Anny Yuniarti, S.Kom, M.Comp.Sc
(NIP. 198106222005012002) (Pembimbing 2)

**SURABAYA
DESEMBER, 2016**

(Halaman ini sengaja dikosongkan)

IMPLEMENTASI K-MEANS BERBASIS GRAPH UNTUK PENGELOMPOKAN PENELITI PADA SISTEM REPOSITORY RISET ITS

Nama Mahasiswa : HIFNIE BILFASH
NRP : 5113100091
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Diana Purwitasari, S.Kom., M.Sc.
Dosen Pembimbing 2 : Anny Yuniarti, S.Kom, M.Comp.Sc

Abstrak

Saat ini ITS sudah memiliki Sistem Repositori Peneliti (resits.its.ac.id) selanjutnya disebut resits. Resits merupakan sistem informasi yang secara khusus menyediakan informasi kepada masyarakat seputar dunia penelitian yang ada di ITS. Pada resits terdapat graf kerjasama peneliti ITS yang hubungan antar grafnya merupakan pengolahan kemiripan topik antar peneliti.

Graf kerjasama peneliti yang dimiliki resits dapat di terapkan metode clustering, untuk mendapatkan fokus topik penelitian per cluster. Metode clustering yang digunakan adalah K-Means berbasis graf. Langkah pertama dari K-Means berbasis graf adalah membentuk minimum spanning tree dari graf kerjasama peneliti ITS, setelah itu tree yang terbentuk diterapkan threshold untuk mendapatkan initial centroid dari cluster. Selanjutnya initial centroid yang terbentuk digunakan untuk mengelompokkan peneliti yang belum memiliki cluster dengan melakukan traverse tree ke initial centroid terdekat. Disamping itu, sebagai pembanding metode cluster kerjasama dengan K-Means berbasis graf. Pada tugas akhir ini juga dikembangkan pembentukan graph kerja sama dengan ekspansi sinonim kata. Sinonim tersebut nantinya akan digunakan dalam membentuk graph kerja sama peneliti ITS.

Metode K-Means berbasis graf memiliki nilai Dunn Index yang lebih baik dibandingkan K-Means berbasis graf ekspansi

sinonim kata. Hal ini disebabkan ekspansi sinonim kata pada judul penelitian menyebabkan keunikan pada judul penelitian semakin berkurang. Sehingga pembentukan graf kerjasama ekspansi sinonim kata tidak menghasilkan cluster graf yang lebih baik daripada graf cluster kerjasama peneliti tanpa ekspansi sinonim kata.

Kata kunci: Sistem Repositori Peneliti, Sinonim, Graph, Kerja sama, Teks Processing.

IMPLEMENTATION OF GRAPH BASED K-MEANS CLASSIFICATION FOR ITS RESEARCH REPOSITORY SYSTEMS

Student's Name : HIFNIE BILFASH
Student's ID : 5113100091
Department : Teknik Informatika FTIF-ITS
First Advisor : Diana Purwitasari, S.Kom., M.Sc.
Second Advisor : Anny Yuniarti, S.Kom, M.Comp.Sc

Abstract

Currently ITS already have Research Repository System (resits.its.ac.id) and referred as resits. Resits is an information system that specifically provide information to the public about research in ITS. In resits there is researchers ITS cooperation graph, the relations between vertex(researchers) is topic similarity between researchers.

Researcher cooperation graf can be applied to clustering methods, the purpose is to get the focus topics per cluster. Clustering method used in this final project is graph based K-Means. The first step of the graph based K-Means is forming a minimum spanning tree from ITS research cooperation graph, then the tree that formed applied threshold value to obtain initial centroid of the cluster. Initial centroid subsequently formed is used to classify researchers who do not have cluster by traversing the tree to the nearest centroid. In addition, as a comparison method of forming the cooperation of researchers graph described above. In this final project also developed the establishment graph of cooperation with expand synonym of a word. The synonyms will be used in the form of ITS research cooperation graph.

Graph based K-Means method have better Dunn Index values than graph based K-Means expand synonym. this is due to the expansion synonym for the title of research led to reduces the

uniqueness of the research title. So that graph based K-Means expand synonym method not produce better graph cluster than graph based K-Means without expand synonym.

Keywords : Research Repository System, Synonym, Graph, Colaboration, Text Processing

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah Swt yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“IMPLEMENTASI K-MEANS BERBASIS GRAPH UNTUK PENGELOMPOKAN PENELITI PADA SISTEM REPOSITORY RISET ITS”

yang merupakan salah satu syarat dalam menempuh ujian sidang guna memperoleh gelar Sarjana Komputer. Selesaiannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT atas izin dan rahmat Nya tugas akhir ini dapat diselesaikan.
2. Keluarga tercinta, Abi, Umi, Nabila, dan Risna Ayu Hestika Prasajo yang telah memberikan dukungan, doa, motivasi, dan perhatian yang luar biasa tanpa henti selama pengerjaan mengerjakan Tugas Akhir ini.
3. Ibu Diana Purwitasari, S.Kom., M.Sc. dan Ibu Anny Yuniarti, S.Kom, M.Comp.Sc. selaku pembimbing I dan II yang telah membimbing dan membantu penulis serta memberikan motivasi dalam menyelesaikan Tugas Akhir ini.
4. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, Ibu Bilqish Amalia S.Kom, M.Kom. selaku dosen wali, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya serta staf karyawan Jurusan Teknik Informatika ITS yang telah memberikan bantuan demi kelancaran administrasi penulis selama kuliah.

5. Risna Ayu Hestika Prasajo yang senantiasa sabar dan memberikan dukungan, doa dan motivasi dalam pengerjaan tugas akhir ini.
 6. Mardiana Sekarsari, Raras Anggita, dan Abyan Faris yang telah menemani, menghibur, memberikan support penuh, dan menjadi sahabat tinggi dan membahagiakan penulis disaat penulis mengalami masa-masa sulit dalam menjalani pedih dan kerasnya kuliah di ITS.
 7. Demsy Iman, Luffi A. S., Yohana Desi, Annisa Trisnawati yang selalu memberikan support berupa dukungan penuh saat melakukan kerja praktek.
 8. Teman-teman Pengurus Harian HMTK Kabinet Optimasi dan teman-teman angkatan 2013 yang telah berjuang bersama menjalani perkuliahan serta menjadi keluarga kedua bagi penulis.
 9. Teman-teman seperjuangan RMK KCV, yang telah menemani dan menyemangati penulis.
 10. Teman-teman administrator KCV, administrator Mobile Innovation Studio, dan rekan-rekan kerja di Direktorat Pengembangan Teknologi Sistem Informasi ITS yang telah menemani dan menyemangati penulis selama penulis menjadi mahasiswa, menjadi rumah kedua penulis selama penulis berkuliah.
 11. Sahabat penulis yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu dan berjuang bersama-sama penulis.
- Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan sehingga dengan kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Desember 2016

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak.....	vii
Abstract	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xxvii
DAFTAR KODE SUMBER	xxix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Permasalahan	3
1.4 Tujuan dan Manfaat	3
1.5 Metodologi	4
1.6 Sistematika Penulisan Laporan	6
BAB II TINJAUAN PUSTAKA.....	9
2.1 Praproses Data.....	9
2.2 K-Means berbasis graph.....	14
2.3 Kamus Bahasa Kateglo Dataset	22
2.4 Cluster Analisis (<i>Dunn Index</i>).....	24
2.5 Betweenness Centrality	26
BAB III ANALISIS DAN PERANCANGAN SISTEM	27
3.1 Analisis.....	27
3.1.1 Analisis Permasalahan	27
3.1.2 Deskripsi Umum Sistem	28
3.2 Perancangan	28
3.2.1 Bahasa Pemrograman Python	28
3.2.2 Pembentukan Cluster Kerjasama Peneliti.....	31
3.2.3 Pembentukan Sinonim Thesaurus Kateglo	43
3.2.4 Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus	53
BAB IV IMPLEMENTASI.....	61
4.1 Lingkungan Implementasi.....	61
4.1.1 Perangkat Keras	61

4.1.2	Perangkat Lunak	61
4.2	Implementasi Basis data.....	62
4.2.1	Daftar Stop Word.....	62
4.2.2	Basis Data Repositori Peneliti ITS	63
4.2.3	Kateglo.....	65
4.3	Implementasi K-Means Berbasis Graph.....	66
4.3.1	Inisiasi Edge Value	67
4.3.2	Minimum Spanning Tree	68
4.3.3	Penerapan Prim Trajectory	69
4.3.4	Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	72
4.4	Implementasi Pembentukan Sinonim Thesaurus Kateglo	74
4.4.1	Praproses Data	74
4.4.2	Pembobotan Kata.....	76
4.5	Ekspansi Pembentukan Graph Kerjasama dengan Sinonim Thesaurus.	81
4.5.1	Data Peneliti dan Judul Publikasi	81
4.5.2	Modifikasi TF x IDF.....	82
BAB V HASIL UJI COBA DAN EVALUASI		89
5.1	Deskripsi Uji Coba	89
5.1.1	Lingkungan Uji Coba.....	89
5.1.2	Data Uji Coba	90
5.1.3	Kuesioner Uji Coba	90
5.2	Uji Coba 1	91
5.2.1	Skenario Uji Coba 1.....	91
5.2.2	Hasil Uji Coba 1	92
5.2.3	Analisis dan Evaluasi.....	94
5.3	Uji Coba 2	95
5.3.1	Skenario Uji Coba 2.....	95
5.3.2	Hasil Uji Coba 2	96
5.3.3	Analisis dan Evaluasi.....	97
5.4	Uji Coba 3	98
5.4.1	Skenario Uji Coba 3.....	98
5.4.2	Hasil Uji Coba 3	98

5.4.3	Analisis dan Evaluasi.....	100
5.5	Uji Coba 4	101
5.5.1	Skenario Uji Coba 4.....	101
5.5.2	Hasil Uji Coba 4	102
5.5.3	Analisis dan Evaluasi.....	102
5.6	Uji Coba 5	104
5.6.1	Skenario Uji Coba 5.....	104
5.6.2	Hasil Uji Coba 5	105
5.6.3	Analisis dan Evaluasi.....	104
5.7	Uji Coba 6	108
5.7.1	Skenario Uji Coba 6.....	108
5.7.2	Hasil Uji Coba 6	108
5.7.3	Analisis dan Evaluasi.....	109
5.8	Uji Coba 7	110
5.8.1	Skenario Uji Coba 7.....	110
5.8.2	Hasil Uji Coba 8	110
5.8.3	Analisis dan Evaluasi.....	111
5.9	Uji Coba 8	114
5.9.1	Skenario Uji Coba 8.....	114
5.9.2	Hasil Uji Coba 8	114
5.9.3	Analisis dan Evaluasi.....	116
BAB VI KESIMPULAN DAN SARAN		119
6.1	Kesimpulan.....	119
6.2	Saran.....	120
DAFTAR PUSTAKA		121
LAMPIRAN		123
BIODATA PENULIS		283

DAFTAR GAMBAR

Gambar 2.1 Contoh gambar graph yang belum diterapkan metode clustering K-Means berbasis graph	15
Gambar 2.2 Contoh graph yang telah diterapkan metode MST ..	17
Gambar 2.3. Contoh pembentukan mst dengan algoritma prim..	18
Gambar 2.4. Contoh graph yang telah dilakukan Prim Trajectory	20
Gambar 2.5. Hasil Clustering	21
Gambar 2.6. PDM Database Kateglo	22
Gambar 3.1 Deskripsi Umum Sistem [Main]	29
Gambar 3.2. Tahapan Proses Pembentukan Cluster Kerjasama Peneliti	32
Gambar 3.3 Graph kerjasama peneliti pada Fakultas Teknologi Informasi ITS sebelum di terapkan metode mst	36
Gambar 3.4 Minimum Spanning Tree yang terbentuk dari data graph kerjasama peneliti Fakultas Teknologi Informasi ITS	38
Gambar 3.5 Initial cluster yang terbentuk dari hasil pengolahan minimum spanning tree dengan prim trajectory	40
Gambar 3.6 Graph Kerjasama Setelah Diterapkan Metode K-Means berbasis Graph	42
Gambar 3.7. Struktur tabel pembentukan thesaurus	43
Gambar 3.8 Alur pembentukan sinonim dari database kateglo ..	44
Gambar 3.9. Pembentukan cluster kerjasama peneliti dengan perhitungan sinonim	55
Gambar 4.1 Tabel tran_bobot_dosen_peneliti	63
Gambar 4.2 Tabel tran_publikasi_dosen_tetap	64
Gambar 4.3 Tabel tran_temp_dosen	64
Gambar 4.4 Tabel proc_definition	66
Gambar 4.5 Tabel lexical	66
Gambar 4.6 Hasil Expand Pembentukan Graph Kerjasama dengan Sinonim Thesaurus	85
Gambar 4.7 Hasil Minimum Spanning Tree dari Expand Pembentukan Graph Kerjasama dengan Sinonim Thesaurus	86

Gambar 4.8 Hasil Clustering Expand Pembentukan Graph Kerjasama dengan Sinonim Thesaurus	87
Gambar 5.1 Rangkang Nilai Betweenness Peneliti pada cluster terbesar FTIF	105
Gambar 5.2 Visualisasi Node Peneliti berdasarkan nilai Betweenness	106
Gambar 5.3 Cluster Besar pada Graph Kerjasama Peneliti FTIF	112
Gambar 5.4 Hasil Pemecahan Cluster Besar pada Graph Kerjasama Peneliti FTIF	113
Gambar 8.1 Gambar graf kerjasama peneliti FTIF ITS sebelum diterapkan metode clustering.....	123
Gambar 8.2 Gambar minimum spanning tree graf kerjasama peneliti ITS.....	124
Gambar 8.3 Gambar graf kerjasama peneliti FTIF ITS ekspansi sinonim kata	125
Gambar 8.4 Gambar minimum spanning tree yang terbentuk dari graf kerjasama peneliti ITS ekspansi sinonim kata	126
Gambar 8.5 Gambar graf cluster peneliti ITS dengan K sama dengan 7	127
Gambar 8.6 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 7	128
Gambar 8.7 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 7	129
Gambar 8.8 Gambar graf peneliti FTSP ITS dengan K sama dengan 7	130
Gambar 8.9 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 7	131
Gambar 8.10 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 7	132
Gambar 8.11 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 7.....	133
Gambar 8.12 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 7.....	134

Gambar 8.13 Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 7.....	135
Gambar 8.14 Gambar graf cluster peneliti jurusan Kimia ITS dengan nilai K sama dengan 7.....	136
Gambar 8.15 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 7.....	137
Gambar 8.16 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 7.....	138
Gambar 8.17 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 7.....	139
Gambar 8.18 Gambar graf cluster peneliti jurusan Teknik Kimia dengan nilai K sama dengan 7.....	140
Gambar 8.19 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 7.....	141
Gambar 8.20 Gambar graf cluster peneliti jurusan Teknik Industri ITS dengan nilai K sama dengan 7.....	142
Gambar 8.21 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 7.....	143
Gambar 8.22 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 7...	144
Gambar 8.23 Gambar graf cluster peneliti jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 7.....	145
Gambar 8.24 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 7.....	146
Gambar 8.25 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 7.....	147
Gambar 8.26 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 7.....	148
Gambar 8.27 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 7.....	149
Gambar 8.28 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 7.....	150
Gambar 8.29 Gambar graf cluster peneliti jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 7.....	151

Gambar 8.30 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 7	152
Gambar 8.31 Gambar graf cluster peneliti jurusan Desain Interior ITS dengan nilai K sama dengan 7.....	153
Gambar 8.32 Gambar graf cluster peneliti jurusan Teknik Perkapalan ITS dengan nilai K sama dengan 7	154
Gambar 8.33 Gambar graf cluster peneliti jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 7	155
Gambar 8.34 Gambar graf cluster peneliti jurusan Teknik Kelautan ITS dengan nilai K sama dengan 7.....	156
Gambar 8.35 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 7	157
Gambar 8.36 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 7.....	158
Gambar 8.37 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 7.....	159
Gambar 8.38 Gambar graf cluster peneliti ITS dengan K sama dengan 10	160
Gambar 8.39 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 10	161
Gambar 8.40 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 10	162
Gambar 8.41 Gambar graf peneliti FTSP ITS dengan K sama dengan 10	163
Gambar 8.42 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 10	164
Gambar 8.43 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 10	165
Gambar 8.44 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 10.....	166
Gambar 8.45 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 10.....	167
Gambar 8.46 Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 10.....	168

Gambar 8.47 Gambar graf clsuter peneliti jurusan Kimia ITS dengan nilai K sama dengan 10.....	169
Gambar 8.48 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 10.....	170
Gambar 8.49 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 10.....	171
Gambar 8.50 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 10.....	172
Gambar 8.51 Gambar graf cluster peneliti jurusan Teknik Kimia dengan nilai K sama dengan 10.....	173
Gambar 8.52 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 10.....	174
Gambar 8.53 Gambar graf cluster peneliti jurusan Teknik Industri ITS dengan nilai K sama dengan 10.....	175
Gambar 8.54 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 10	176
Gambar 8.55 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 10.	177
Gambar 8.56 Gambar graf cluster peneliti jurusan Managemen Bisnis ITS dengan nilai K sama dengan 10.....	178
Gambar 8.57 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 10.....	179
Gambar 8.58 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 10.....	180
Gambar 8.59 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 10	181
Gambar 8.60 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 10.....	182
Gambar 8.61 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 10	183
Gambar 8.62 Gambar graf cluster peneliti jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 10.....	184
Gambar 8.63 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 10	185

Gambar 8.64 Gambar graf cluster peneliti jurusan Desain Interior ITS dengan nilai K sama dengan 10.....	186
Gambar 8.65 Gambar graf cluster peneliti jurusan Teknik Perkapalan ITS dengan nilai K sama dengan 10	187
Gambar 8.66 Gambar graf cluster peneliti jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 10	188
Gambar 8.67 Gambar graf cluster peneliti jurusan Teknik Kelautan ITS dengan nilai K sama dengan 10.....	189
Gambar 8.68 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 10	190
Gambar 8.69 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 10.....	191
Gambar 8.70 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 10.....	192
Gambar 8.71 Gambar graf cluster peneliti ITS dengan K sama dengan 16	193
Gambar 8.72 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 16	194
Gambar 8.73 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 16	195
Gambar 8.74 Gambar graf cluster peneliti FTSP ITS dengan K sama dengan 16	196
Gambar 8.75 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 16	197
Gambar 8.76 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 16	198
Gambar 8.77 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 16.....	199
Gambar 8.78 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 16.....	200
Gambar 8.79 Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 16.....	201
Gambar 8.80 Gambar graf cluster peneliti jurusan Kimia ITS dengan nilai K sama dengan 16.....	202

Gambar 8.81 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 16.....	203
Gambar 8.82 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 16.....	204
Gambar 8.83 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 16.....	205
Gambar 8.84 Gambar graf cluster peneliti jurusan Teknik Kimia dengan nilai K sama dengan 16.....	206
Gambar 8.85 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 16.....	207
Gambar 8.86 Gambar graf cluster peneliti jurusan Teknik Industri ITS dengan nilai K sama dengan 16.....	208
Gambar 8.87 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 16	209
Gambar 8.88 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 16.	210
Gambar 8.89 Gambar graf cluster peneliti jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 16.....	211
Gambar 8.90 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 16.....	212
Gambar 8.91 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 16.....	213
Gambar 8.92 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 16	214
Gambar 8.93 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 16.....	215
Gambar 8.94 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 16	216
Gambar 8.95 Gambar graf cluster peneliti jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 16.....	217
Gambar 8.96 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 16	218
Gambar 8.97 Gambar graf cluster peneliti jurusan Desain Interior ITS dengan nilai K sama dengan 16.....	219

Gambar 8.98 Gambar graf cluster peneliti jurusan Teknik Perkapalan ITS dengan nilai K sama dengan 16	220
Gambar 8.99 Gambar graf cluster peneliti jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 16	221
Gambar 8.100 Gambar graf cluster peneliti jurusan Teknik Kelautan ITS dengan nilai K sama dengan 16	222
Gambar 8.101 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 16	223
Gambar 8.102 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 16	224
Gambar 8.103 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 16	225
Gambar 8.104 Gambar graf cluster peneliti ITS ekspansi sinonim kata dengan K sama dengan 10	226
Gambar 8.105 Gambar graf cluster peneliti ekspansi sinonim kata FMIPA ITS dengan K sama dengan 10	227
Gambar 8.106 Gambar graf cluster peneliti ekspansi sinonim kata FTI ITS dengan K sama dengan 10	228
Gambar 8.107 Gambar graf peneliti ekspansi sinonim kata FTSP ITS dengan K sama dengan 10	229
Gambar 8.108 Gambar graf cluster peneliti ekspansi sinonim kata FTK ITS dengan nilai K sama dengan 10	230
Gambar 8.109 Gambar graf cluster peneliti ekspansi sinonim kata FTIF ITS dengan nilai K sama dengan 10	231
Gambar 8.110 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Fisika ITS dengan nilai K sama dengan 10	232
Gambar 8.111 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Matematika ITS dengan nilai K sama dengan 10	233
Gambar 8.112 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Statistika ITS dengan nilai K sama dengan 10	234
Gambar 8.113 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Kimia ITS dengan nilai K sama dengan 10	235
Gambar 8.114 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Biologi ITS dengan nilai K sama dengan 10	236

Gambar 8.115 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Mesin ITS dengan nilai K sama dengan 10.....	237
Gambar 8.116 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Elektro ITS dengan nilai K sama dengan 10....	238
Gambar 8.117 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Kimia dengan nilai K sama dengan 10.....	239
Gambar 8.118 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Fisika dengan nilai K sama dengan 10.....	240
Gambar 8.119 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Industri ITS dengan nilai K sama dengan 10 ...	241
Gambar 8.120 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 10	242
Gambar 8.121 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 10	243
Gambar 8.122 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 10	244
Gambar 8.123 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Sipil ITS dengan nilai K sama dengan 10	245
Gambar 8.124 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Arsitektur ITS dengan nilai K sama dengan 10.....	246
Gambar 8.125 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 10	247
Gambar 8.126 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Desain Produk ITS dengan nilai K sama dengan 10	248
Gambar 8.127 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Geomatika ITS dengan nilai K sama dengan 10	249
Gambar 8.128 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 10	250

Gambar 8.129 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Geofisika ITS dengan nilai K sama dengan 10	251
Gambar 8.130 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Desain Interior ITS dengan nilai K sama dengan 10....	252
Gambar 8.131 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Perkapalan ITS dengan nilai K sama dengan 10	253
Gambar 8.132 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 10	254
Gambar 8.133 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Kelautan ITS dengan nilai K sama dengan 10 .	255
Gambar 8.134 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Transportasi Laut ITS dengan nilai K sama dengan 10	256
Gambar 8.135 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Informatika ITS dengan nilai K sama dengan 10	257
Gambar 8.136 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Sistem Informasi ITS dengan nilai K sama dengan 10.	258
Gambar 8.137 Gambar cluster kerjasama yang terpecah dalam graf cluster kerjasama peneliti FTIF ITS	259
Gambar 8.138 Gambar cluster kerjasama keseluruhan dalam graf kerjasama peneliti ITS.....	260
Gambar 8.139 Gambar cluster kerjasama yang terpecah dalam graf cluster kerjasama peneliti FTIF ITS	261
Gambar 8.140 Gambar cluster kerjasama keseluruhan dalam graf kerjasama peneliti ITS.....	262
Gambar 8.141 Keterkaitan antar jurusan yang memiliki kemiripan fokus penelitian	263

DAFTAR TABEL

Tabel 2.1. Contoh Menghitung TF	11
Tabel 2.2. Contoh Menghitung IDF	13
Tabel 2.3. Contoh Menghitung TF x IDF	14
Tabel 2.4. Tabel Detail Database Kateglo.....	23
Tabel 2.5 Tabel Contoh Kata Berdasarkan Tipe Leksikal.....	24
Tabel 3.1 Data Kerjasama Peneliti ITS	33
Tabel 3.2 Data Kerjasama Peneliti ITS dalam bentuk nilai jarak edge	34
Tabel 3.3. Keterangan database pembentukan thesaurus	45
Tabel 3.4 Tabel publikasi peneliti dosen ITS.....	54
Tabel 3.5 Tabel distinct publikasi peneliti dosen ITS	56
Tabel 4.1 Tabel keterangan stopword	62
Tabel 4.2 Keterangan tabel hasil duplikasi data dari basis data peneliti ITS.....	65
Tabel 4.3 Tabel Pengubahan Nilai Konstanta C dan K.....	71
Tabel 5.1 Tabel Penghitungan Nilai Dunn Index dari Masing-masing Data Cluster Kerjasama Peneliti	92
Tabel 5.2 Tabel Hasil Uji Coba 2.....	96
Tabel 5.3 Tabel Hasil Uji Coba 3.....	98
Tabel 5.4 Tabel Hasil Ujicoba 4.....	102
Tabel 5.5 Tabel Hasil Uji Coba 6.....	108
Tabel 5.6 Tabel Hasil Uji Coba 7.....	110
Tabel 5.7 Tabel Waktu Eksekusi.....	114
Tabel 5.8 Tabel Jumlah Data yang Dihasilkan.....	115
Tabel 8.1 Tabel daftar 10 peneliti tertinggi yang memiliki term penelitian tidak termasuk dalam sinonim kata	264
Tabel 8.2 Tabel daftar 90 term tertinggi berdasarkan nilai IDF dari judul penelitian yang tidak terdapat pada kamus kata kateglo ..	264
Tabel 8.3 Tabel term topik terbaik yang mewakili dosen FTIF	268

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4-1 Proses pengubahan nilai bobot kerjasama ke nilai edge	67
Kode Sumber 4-2 Proses Pembentukan Minimum Spanning Tree	69
Kode Sumber 4-3 Proses Penerapan Prim Trajectory	71
Kode Sumber 4-4 Proses Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	73
Kode Sumber 4-5 Implementasi Tokenisasi.....	74
Kode Sumber 4-6 Implementasi Penghapusan Tanda Baca Kalimat	75
Kode Sumber 4-7 Implementasi Penghapusan Stop Word	75
Kode Sumber 4-8 Implementasi Stemming Kata.....	76
Kode Sumber 4-9 Penghitungan Term Frequency	77
Kode Sumber 4-10 Penghitungan Inverse Document Frequency.....	77
Kode Sumber 4-11 Fungsi Mendapatkan Nilai IDF.....	78
Kode Sumber 4-12 Fungsi Menghitung nilai TFIDF	78
Kode Sumber 4-13 Fungsi Mendapatkan Nilai TF IDF	79
Kode Sumber 4-14 Penghitungan Cosine Similarity.....	80
Kode Sumber 4-15 Inisiasi data publikasi peneliti.....	81
Kode Sumber 4-16 Mengambil Data Sinonim Dari File Json.....	82
Kode Sumber 4-17 Penghitungan Modifikasi Nilai TF	82
Kode Sumber 4-18 Penghitungan Modifikasi Nilai IDF.....	83

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemerintah memberikan perhatian penuh terutama kepada para kalangan akademisi untuk melakukan penelitian, seperti dukungan dana serta lomba-lomba keilmiah. Kegiatan ekstrakurikuler keilmiah juga dikembangkan mulai pendidikan tingkat menengah hingga perguruan tinggi. Sebagai salah satu perguruan tinggi di Indonesia, Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan para peneliti yang ada di dalamnya aktif memberikan kontribusi terhadap dunia penelitian Indonesia melalui publikasi jurnal dan seminar penelitian secara rutin setiap tahunnya [2]. Para peneliti dalam kapasitasnya sebagai penyedia iptek harus turut serta berperan dalam inovasi nasional. Kegiatan penelitian/riset selama ini sering terjadi antara satu dengan lain tidak ada keterkaitan. Perlu diusahakan agar kegiatan penelitian dapat dilakukan secara holistik, lebih fokus, lebih kontekstual dan ada kerja sama antar-peneliti dalam penentuan topik penelitian [1]. Penelitian yang dilakukan oleh banyak periset Indonesia sering bersifat sektoral dan tidak memiliki keterkaitan interdisipliner dengan penelitian di bidang lain. Padahal, penelitian yang saling mendukung satu dengan lainnya sangatlah penting. Penelitian interdisipliner dapat membantu mengembangkan kebijakan yang tepat sasaran dan efisien. Atas dasar itulah sangat penting untuk mengetahui kerja sama antar peneliti di ITS.

Saat ini ITS sudah memiliki Sistem Repositori Peneliti (resits.its.ac.id) yang selanjutnya disebut *resits*. *Resits* merupakan sistem informasi yang secara khusus memberikan informasi kepada masyarakat seputar dunia penelitian yang ada di ITS. Beberapa fitur yang terdapat dalam sistem tersebut yaitu pengguna dapat melakukan pencarian peneliti dengan kriteria tertentu, melihat daftar publikasi jurnal penelitian terakhir, serta fitur lainnya. Pada sistem informasi tersebut pengguna dapat melakukan pencarian peneliti berdasarkan pengelompokan area peneliti

(fakultas). Pada sistem informasi tersebut juga sudah memiliki visualisasi data kerja sama peneliti dalam bentuk graf yang menarik dan mudah dipahami. Namun representasi model graf yang ditampilkan merupakan hasil pengolahan data melalui *thresholding* saja tanpa dilakukan proses pengelompokan atau *clustering* kerja sama antar peneliti. Sehingga data yang ditampilkan belum dilakukan pengelompokan kerja sama antar peneliti melalui metode *clustering*.

Oleh karena itu dalam tugas akhir ini akan dibuat sebuah modul yang akan menjadi bagian dari fitur Sistem Informasi Repositori Peneliti. Modul yang akan dibuat ini akan berfokus pada *clustering* peta kerja sama antar peneliti berdasarkan topik. Dengan dilakukannya *clustering* pada peta kerja sama berdasarkan topik, data yang akan didapatkan tentunya akan lebih akurat dari sebelumnya. Hal ini dikarenakan data diambil dari hasil penelitian yang pernah dibuat oleh setiap peneliti. Kemudian data yang sudah didapatkan akan divisualisasikan. Dalam tugas akhir ini akan dipilih bentuk visualisasi data peta kerja sama antar peneliti menggunakan pemodelan graf. Setelah itu data tersebut akan dilakukan proses *clustering* untuk pengelompokan kerja sama yang dilakukan peneliti menggunakan metode *K-Means* berbasis graf. Dengan visualisasi pemodelan graf yang telah diblaster, peta kerja sama antar peneliti akan terlihat lebih informatif dan akurat. Misalnya saja untuk melihat pengelompokan peneliti berdasarkan topiknya, bisa dilihat melalui koloni-koloni yang terbentuk dari kumpulan *node/vertex* (titik) yang merepresentasikan setiap peneliti. Kemudian bisa juga dilihat hubungan kerja sama antar peneliti melalui jarak yang direpresentasikan menggunakan *edge* (garis) yang menghubungkan antar *vertex* peneliti, semakin sering antar peneliti bekerja sama maka jarak *vertex*nya pun akan semakin dekat serta sebaliknya.

Harapan yang ingin dicapai dalam tugas akhir ini, para peneliti, mahasiswa ataupun masyarakat umum dapat mengetahui peta kerja sama antar peneliti berdasarkan topik penelitian yang ada di ITS dengan lebih akurat.

1.2 Rumusan Masalah

Tugas akhir ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Bagaimana mengetahui peneliti yang memiliki kesamaan berdasarkan topik dari penelitian yang pernah dibuat?
2. Bagaimana memodelkan peneliti-peneliti yang mirip berdasarkan topik menggunakan visualisasi pemodelan graf berbasis web?
3. Bagaimana mengelompokkan peneliti-peneliti berdasarkan nilai kerja sama pada graf kerja sama antar peneliti?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada tugas akhir ini memiliki batasan sebagai berikut:

1. Menggunakan kamus data resmi Pangkalan Data Perguruan Tinggi ITS.
2. Visualisasi hasil pengelompokan berbasis web.
3. Bahasa pemrograman yang digunakan adalah *PHP* dengan kaskas kerja *Laravel*, *Python*.
4. Menggunakan basis data SQL Server 2008.
5. Data yang digunakan yaitu berasal dari data sekunder (data replika dari data asli) yang sudah disesuaikan dengan standar data primer (data asli) yang diambil dari Lembaga Pengembangan Teknologi dan Sistem Informasi (LPTSI) ITS.

1.4 Tujuan dan Manfaat

1. Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

- a. Merancang dan membangun sebuah Modul yang akan menjadi bagian dari fitur Sistem Repositori Peneliti. Modul tersebut berupa visualisasi peta kerja sama peneliti berdasarkan topik penelitian yang ada di ITS.

menggunakan pemodelan graf *clustering* dengan metode *K-Means Clustering*.

- b. Mengimplementasikan metode *K-Means Clustering* pada pengelompokan data kerja sama peneliti di ITS.
- c. Mengimplementasikan *expand K-Means Clustering* dengan memperhatikan sinonim kata pada data peneliti ITS.
- d. Membentuk tesaurus sinonim bahasa Indonesia dengan menggunakan data set dari kataglo.

2. Manfaat

Memudahkan para peneliti, mahasiswa dan masyarakat umum mengetahui peta kerja sama peneliti yang ada di ITS berdasarkan kemiripan topik penelitian dan sudah dilakukan pengelompokan kerja sama peneliti dengan metode yang ada pada tugas akhir ini.

1.5 Metodologi

Pembuatan tugas akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

A. Penyusunan Proposal

Proposal tugas akhir ini berisi rencana tugas akhir yang akan dikerjakan sebagai syarat untuk menyelesaikan studi dan meraih gelar Strata-1 Teknik Informatika. Terdapat penjelasan mengenai latar belakang pengambilan tema, rumusan masalah, batasan masalah, tujuan, dan manfaat dari rencana tugas akhir ini. Selain itu juga dijelaskan mengenai metode apa saja yang digunakan serta penjelasannya. Agar lebih mudah dipahami, penjelasan disertai dengan diagram alir.

B. Studi Literatur

Pada tahap ini dilakukan untuk mencari informasi dan studi literatur apa saja yang dapat dijadikan referensi untuk membantu pengerjaan Tugas Akhir ini. Informasi didapatkan dari buku dan literatur yang berhubungan dengan metode yang digunakan. Informasi yang dicari adalah *K-Means Clustering berbasis graph*, *Indexing kata-kata standart Information Retrieval*, dan metode-metode *preprocessing Text Mining* seperti *Stemming*, *Punctuation Removal*, *Tokenization*, *Stopword Removal* dan *penyusunan thesaurus menggunakan database kateblo*. Tugas akhir ini juga mengacu pada literatur jurnal karya Laurent Galluccio , Olivier Michel, Pierre Comon, dan Alfred O. Hero dengan judul “*Graph Based K-Means Clustering*” yang diterbitkan pada tahun 2012.

C. Implementasi Sistem

Implementasi merupakan tahap untuk membangun metode-metode yang sudah diajukan pada proposal Tugas Akhir. Untuk membangun algoritma yang telah dirancang sebelumnya, maka dilakukan implementasi dengan menggunakan suatu perangkat lunak yaitu *PyCharm* dan *PHPStorm* sebagai IDE, *Navicat* sebagai *Database Client*, dan basis data *SQLServer 2008*.

D. Pengujian dan Evaluasi

Pada tahap ini algoritma yang telah disusun diuji coba dengan menggunakan data uji coba yang ada. Data uji coba tersebut diuji coba dengan menggunakan suatu perangkat lunak dengan tujuan mengetahui kemampuan metode yang digunakan dan mengevaluasi hasil tugas akhir dengan jurnal pendukung yang ada. Hasil evaluasi juga mencakup pengelompokan data peneliti ITS.

E. Penyusunan Buku

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan tugas akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.6 Sistematika Penulisan Laporan

Buku tugas akhir ini disusun dengan tujuan untuk memberikan gambaran tentang pengerjaan tugas akhir yang telah dilakukan. Buku tugas akhir ini terbagi menjadi enam bab, yaitu:

A. Bab I. Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan tugas akhir.

B. Bab II. Tinjauan Pustaka

Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang referensi pemrosesan teks yang akan dipakai dalam pembentukan graph peneliti dan metode K-Means berbasis graph serta pembentukan thesaurus.

C. Bab III. Perancangan dan Analisis Sistem

Bab ini berisi pembahasan mengenai perancangan dari metode K-Means berbasis graph dan penambahan thesaurus sinonim pada K-Means berbasis graph yang digunakan untuk pengelompokan data peneliti di ITS.

D. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk *Pseudocode* dari metode yang digunakan dalam pembuatan sistem.

E. Bab V. Hasil Uji Coba dan Evaluasi

Bab ini berisikan skenario uji coba, hasil uji coba, analisis, dan evaluasi dari pengelompokan data peneliti ITS menggunakan *Graph Based K-Means*. Uji coba dilakukan dengan menggunakan dataset peneliti pada Sistem Repositori Peneliti ITS.

F. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir, dan saran untuk pengembangan solusi ke depannya.

G. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam tugas akhir. Teori-teori tersebut diantaranya adalah *exhaustive search*, *autocorrelation*, dan beberapa teori lain yang mendukung pembuatan tugas akhir.

2.1 Praproses Data

Algoritma pemrosesan teks digunakan untuk melakukan ekstraksi kata pada kumpulan dokumen dan pembentukan sinonim dari kata hasil ekstraksi. Algoritma ini bisa disebut sebagai tahapan praproses data.

A. Tokenisasi

Tokenisasi adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token/bagian-bagian tertentu [3]. Sebagai contoh, tokenisasi dari kalimat "Sistem sensor pengukuran minyak berdasarkan pada pengukuran kapasitansi dan panjang berkas pembiasan cahaya" menghasilkan tiga belas token, yakni: "Sistem", "sensor", "pengukuran", "minyak", "berdasarkan", "pada", "pengukuran", "kapasitansi", "dan", "panjang", "berkas", "pembiasan", "cahaya". Biasanya, yang menjadi acuan pemisah antar token adalah spasi dan tanda baca. Tokenisasi sering kali dipakai dalam ilmu linguistik dan hasil tokenisasi berguna untuk analisis teks lebih lanjut. Contoh program tokenisasi yang dapat diakses via Online adalah *MorphAdorner* dan *NLTK Tokenizer* [4].

B. Penghapusan Stop word

Stop words adalah kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Oleh karena itu, kata-kata yang tergolong dalam *stop word*

harus dihapus dalam susunan token untuk meningkat kualitas dari perhitungan selanjutnya. Contoh kata-kata yang termasuk kedalam stop word adalah : ada, adalah, adanya, adapun, agak, agaknya, dst [5]. Sehingga token seperti contoh diatas akan berubah menjadi "Sistem", "sensor", "pengukuran", "minyak", "berdasarkan", "pengukuran", "kapasitansi", "panjang", "berkas", "pembiasaan", "cahaya" karena kata "pada", "dan" merupakan stop word dan sudah dihilangkan dari susunan kata dalam token [4].

C. Penghapusan Tanda Baca

Removing punctuation adalah salah satu metode preproses pengolahan teks dengan menghapus semua tanda baca dalam kata. Sehingga kata yang akan diproses hanya akan berisi karakter alphabet tanpa tanda baca. Contoh token "selesai.", "sekarang!" akan berubah menjadi "selesai", "sekarang" [4].

D. Proses Stemming

Stemming merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Proses *stemming* dilakukan dengan menghilangkan semua imbuhan (afiks) baik yang terdiri dari awalan (prefiks) sisipan (infiks) maupun akhiran (sufiks) dan kombinasi dari awalan dan akhiran (konfiks) [4].

Contoh term "menahan" akan menghasilkan tahanan dan term "berbalas-balasan" akan berubah menjadi "balas". Pustaka online yang bisa dipakai saat ini untuk bahasa indonesia adalah sastrawi.

E. Penghitungan Term Frequency (TF)

Tf menyatakan jumlah berapa banyak keberadaan suatu term dalam satu dokumen dan kemudian dilogaritmikan agar mengurangi besarnya bilangan, dimana logaritmik suatu bilangan akan mengurangi digit jumlah, misalnya 1000 dengan $\log(1000)$

hanya menghasilkan angka tiga [4]. Rumus Tf sebagai mana ditunjukkan pada Persamaan 2.1:

$$TF_{t,d} = \begin{cases} 1 + \log_{10} f_{t,d} & \text{if } f_{t,d} > 0 \\ 0, & \text{if } f_{t,d} = 0 \end{cases}$$

dimana :

t = term

d = dokumen

$f_{t,d}$ = frekuensi term t pada dokumen d

$TF_{t,d}$ = nilai TF dari term t pada dokumen d

Persamaan 2.1 Rumus penghitungan TF

Jadi jika suatu term terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = $1 + \log(5) = 1.699$. Tetapi jika term tidak terdapat dalam dokumen tersebut, bobotnya adalah nol.

Contoh kalimat 1 “Sistem sensor pengukuran minyak berdasarkan pada panjang berkas pembiasan cahaya” dan kalimat 2 “Sistem pengukuran kedalaman laut dengan pergerakan ubur-ubur” dianggap sebagai dokumen dan setiap kata adalah term maka didapat perhitungan sesuai dengan Tabel 2.1:

Tabel 2.1. Contoh Menghitung TF

Term	F_{d1}	TF_{d1}	F_{d2}	TF_{d2}
Sistem	1	1	1	1
Sensor	1	1	0	0
Pengukuran	2	1.3010299	1	1
Minyak	2	1.3010299	0	0
Berdasarkan	1	1	0	0
Berkas	2	1.3010299	0	0
Panjang	1	1	0	0
Pembiasan	1	1	0	0

Cahaya	1	1	0	0
Kedalaman	0	0	1	1
Laut	0	0	1	1
Pergerakan	0	0	1	1
Ubur-ubur	0	0	1	1

F. Penghitungan Invers Document Frequency (IDF)

Terkadang suatu term muncul di hampir sebagian besar dokumen mengakibatkan proses pencarian term unik terganggu. Idf berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar di seluruh koleksi dokumen kita [4]. Rumusnya adalah dengan inverse document frequency. Document frequency adalah seberapa banyak suatu term muncul di seluruh document yang diselidiki. Rumus penghitungan IDF terdapat pada Persamaan 2.2.

$$idf_t = \log_{10} \left(\frac{N + 1}{df_t} \right)$$

dimana :

t = term

N = total dokumen

df_t = frekuensi kemunculan term t pada seluruh dokumen

idf_t = nilai IDF dari term t

Persamaan 2.2. Rumus penghitungan IDF

Sehingga bobot akhir suatu term adalah dengan mengalikan keduanya yaitu $tf \times idf$. Berikut ini kita mengambil contoh suatu kasus. Misalnya kita memiliki kalimat 1 “Sistem sensor pengukuran minyak berdasarkan pada panjang berkas pembiasan cahaya” dan kalimat 2 “Sistem pengukuran kedalaman

laut dengan pergerakan ubur-ubur” dianggap sebagai dokumen dan setiap kata adalah term maka didapat sesuai dengan Tabel 2.1:

Tabel 2.2. Contoh Menghitung IDF

Term	DF	IDF
Sistem	2	0.17609
Sensor	1	0.47712
Pengukuran	2	0.17609
Minyak	1	0.47712
Berdasarkan	1	0.47712
Berkas	1	0.47712
Panjang	1	0.47712
Pembiasaan	1	0.47712
Cahaya	1	0.47712
Kedalaman	1	0.47712
Laut	1	0.47712
Pergerakan	1	0.47712
Ubur-ubur	1	0.47712

G. Penghitungan TF x IDF

Untuk mendapatkan bobot akhir dari suatu term maka dilakukan pengalihan antara TF dan IDF [4]. Contoh penghitungan TF x IDF menggunakan nilai dari TF dan IDF pada Tabel 2.1 dan Tabel 2.2 terdapat pada Tabel 2.3. Berikut Persamaan 2.1 untuk menghitung nilai TF IDF:

$$w_{dt} = TF_{dt} \times IDF_t$$

dimana :

t = term

d = dokumen

$TF_{t,d}$ = nilai TF dari term t pada dokumen d

idf_t = nilai IDF dari term t

Persamaan 2.3. Rumus Penghitungan TF-IDF

Tabel 2.3. Contoh Menghitung TF x IDF

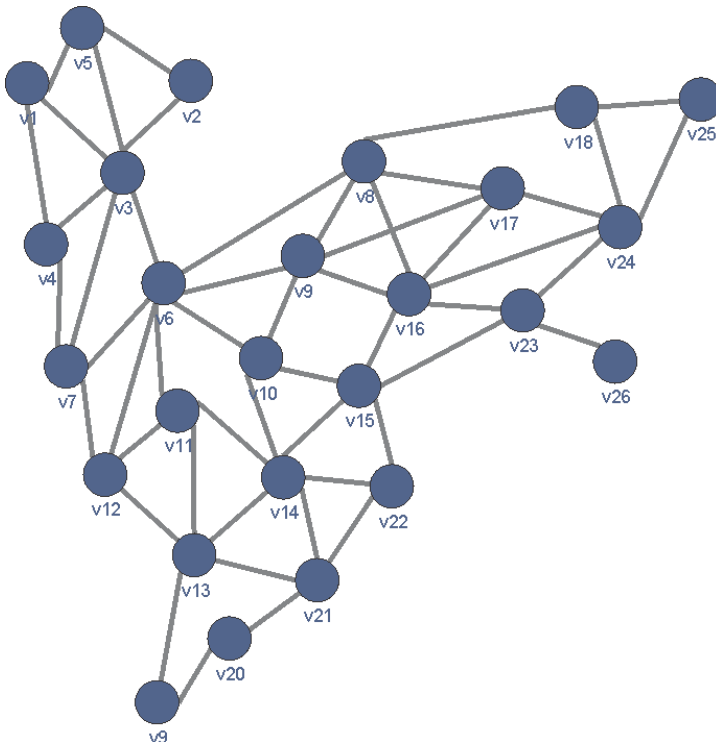
Term	TF _{d1}	TF _{d2}	IDF	W _{d1}	W _{d2}
Sistem	1	1	0.17609	0.17609	0.17609
Sensor	1	0	0.47712	0.47712	0
Pengukuran	1.3010299	1	0.17609	0.22909	0.17609
Minyak	1.3010299	0	0.47712	0.62456	0
Berdasarkan	1	0	0.47712	0.47712	0
Berkas	1.3010299	0	0.47712	0.62456	0
Panjang	1	0	0.47712	0.47712	0
Pembiasaan	1	0	0.47712	0.47712	0
Cahaya	1	0	0.47712	0.47712	0
Kedalaman	0	1	0.47712	0	0.47712
Laut	0	1	0.47712	0	0.47712
Pergerakan	0	1	0.47712	0	0.47712
Ubur-ubur	0	1	0.47712	0	0.47712

2.2 K-Means berbasis graph

Pada dasarnya tujuan melakukan clustering adalah untuk mendapat suatu informasi dari sebuah kumpulan data hingga mendapat suatu kelompok-kelompok data atau cluster. Gambar 2.1 merupakan graf yang belum diterapkan metode clustering. Kebanyakan metode clustering diterapkan pada data yang memiliki dimensi atau nilai pada bidang koordinat, beda halnya dengan data graph, clustering tidak dapat dilakukan pada data graph karena data yang dipakai berupa data dengan fitur berupa informasi sebuah edge. Nilai edge terdiri dari 3 komponen utama yaitu vertex asal, vertex tujuan dan nilai dari edge itu sendiri. Oleh karena itu pada kasus clustering graph perlu dilakukan modifikasi pada metode clustering pada umumnya.

K-Means adalah sebuah metode clustering yang memiliki dua fase yaitu fase inisialisasi centroid awal dan menghitung nilai kedekatan suatu titik dengan centroid initial terdekat hingga cluster yang dihasilkan tidak memiliki perubahan anggota. Pada K-Means

berbasis graph ini memiliki kesamaan dengan K-Means pada umumnya namun metode didalamnya memiliki sedikit modifikasi. Pada K-Means berbasis graph untuk menentukan initial centroid dilakukan dengan melakukan metode pembentukan minimum spanning tree, lalu dari mst yang terbentuk dilakukan thresholding untuk mendapatkan nilai K, selanjutnya untuk menghitung kedekatan antar vertex ke cluster initial maka dilakukan graph traversal pada masing-masing mst yang terbentuk hingga semua vertex memiliki cluster. berikut penjelasan lebih lengkap mengenai K-Means clustering berbasis graph [6].



Gambar 2.1 Contoh gambar graph yang belum diterapkan metode clustering K-Means berbasis graph

A. Proses Pembentukan Minimum Spanning Tree

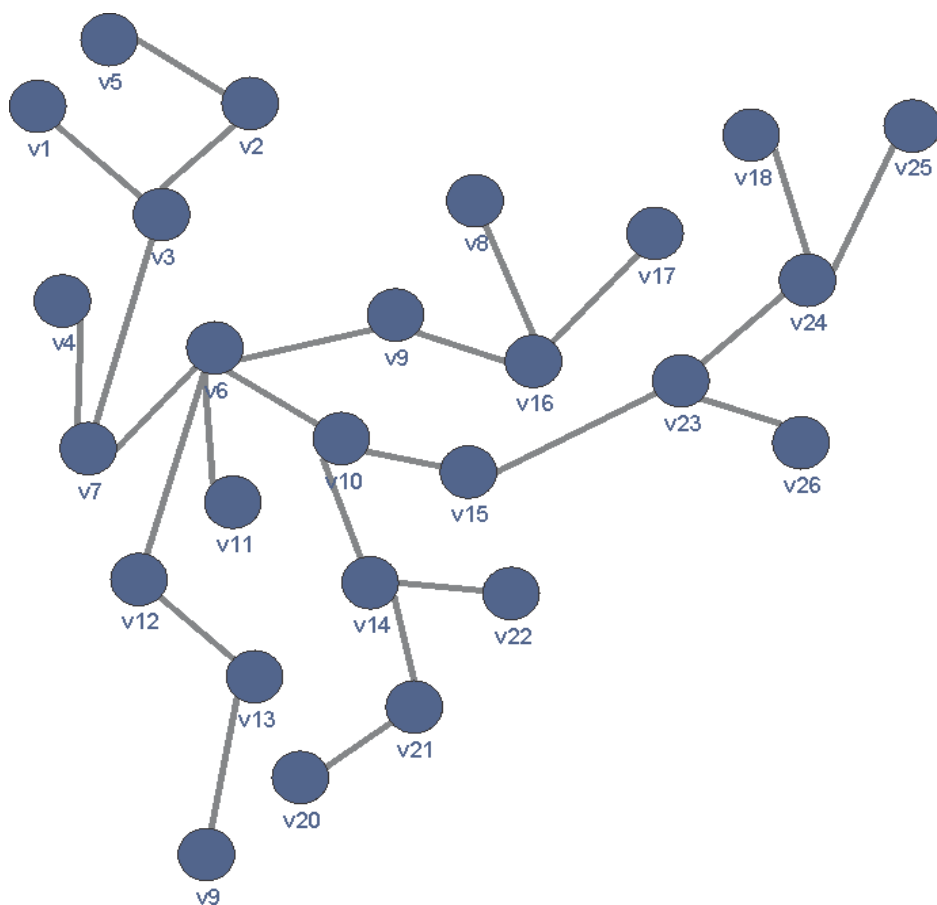
Minimum Spanning Tree pada tugas akhir ini digunakan untuk mendapatkan bentuk *undirected acyclic graph* yang sebelumnya *undirected cyclic graph*. *undirected acyclic graph* merupakan sebuah graph yang tidak memiliki arah dan tidak terjadi loop pada suatu subgraphnya. Sehingga hasil dari Minimum Spanning Tree adalah sebuah tree dari dataset diatas sesuai dengan Gambar 2.2. Minimum Spanning Tree pada dasarnya menerapkan metode umum graph cutting yang akan menghapus edge yang memiliki nilai tidak optimal dalam pembentukan tree seperti pada Gambar 2.3. Pseudocode pembentukan minimum spanning tree terdapat pada Pseudocode 2.1.

Input : graph G dengan vertex V dan edge E
 Output : graph MST yang merupakan minimum spanning tree

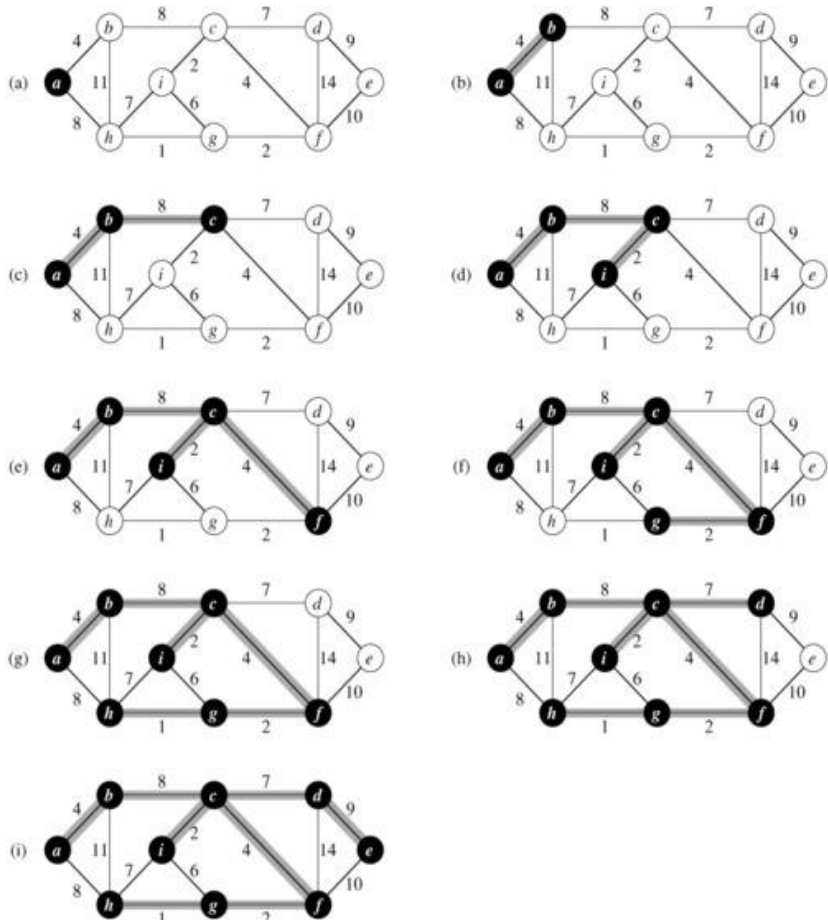
Pseudocode :

1. Inisiasi graph MST yang akan menjadi minimum spanning tree pada akhir proses dengan memilih acak salah satu V sebagai initial tree (V_i).
2. Inisiasi sebuah priority queue Q dengan semua E yang dimiliki V_i pada langkah 1.
3. Ulang hingga Q kosong
4. Jika ada salah satu dari kedua vertex pada $Q.front()$ yang belum dikunjungi
5. MST.append($Q.front()$)
6. Untuk setiap tmpE yang dimiliki vertex pada $Q.front()$
7. Jika ada salah satu dari kedua vertex pada tmpE yang belum dikunjungi
8. Tandai kedua vertex pada $Q.front()$ sudah di kunjungi
9. Keluarkan $Q.front()$ dari priority queue $Q.pop()$
10. Return MST

Pseudocode 2.1 Pseudocode pembentukan MST



Gambar 2.2 Contoh graph yang telah diterapkan metode MST



Gambar 2.3. Contoh pembentukan mst dengan algoritma prim

B. Penerapan Prim Trajectory pada Minimum Spanning Tree

Pada tahap ini merupakan tahap penyempurnaan dari Minimum Spanning Tree. Karena metode Minimum Spanning Tree rentan menghasilkan noise dan outliers pada dataset. Metode

Prim Trajectory mengolah skema Minimum Spanning Tree untuk menghasilkan kandidat subset dari vertex yang merupakan representasi mode dari kerapatan graph. Sehingga vertex-vertex yang terpisah dari yang lain (outlier) akan dihapus dari Minimum Spanning Tree. Pada Prim Trajectory ini juga dilakukan Thresholding. Thresholding merupakan metode untuk menentukan suatu properti dari atribut dengan membandingkan nilai pada suatu konstanta. Thresholding disini digunakan untuk menentukan penghapusan vertex pada Minimum Spanning Tree dalam proses Prim Trajectory [6]. Untuk penentuan nilai dari threshold akan menggunakan threshold hasil percobaan dari Sistem Repositori Peneliti ITS. Gambar 2.4 merupakan gambar hasil penerapan prim trajectory untuk mendapatkan cluster awal, cluster awal tersebut akan menjadi cikal bakal masing-masing cluster. Pseudocode untuk metode prim trajectory sesuai dengan Pseudocode 2.2:

Input : Graph minimum spanning tree MST dengan vertex V dan edge E , konstanta M untuk threshold nilai edge dan konstanta N untuk mendapatkan jumlah N pada proses selanjutnya

Output : initial cluster C_i (innercluster)

Pseudocode :

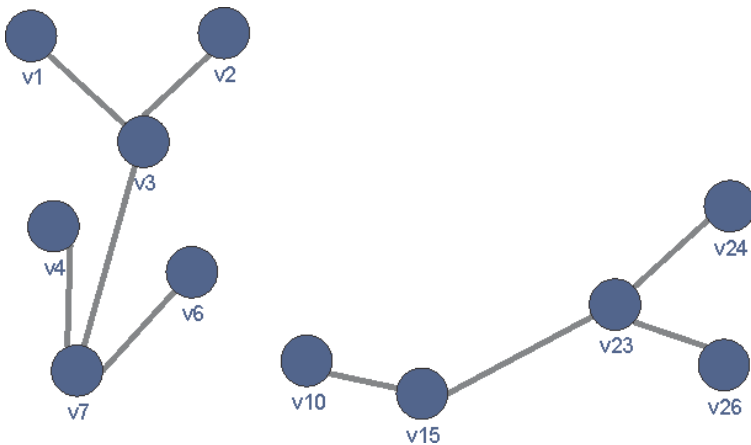
```

1.  $E = E < M$ 
2. Sort( $E$ ) urutan menaik
3. Ulang untuk semua  $E$ 
4.    $tmpE = E$ 
5.   Jika ada salah satu dari kedua vertex
      pada  $tmpE$  yang belum dikunjungi
6.     Tandai kedua vertex sudah dikunjungi
7.      $neighborE = E$  yang terhubung dengan
       $tmpE$  dengan nilai  $< M$ 

8.   Jika  $count(neighborE) > N$ 
9.      $C_i.append(neighborE)$ 

```

Pseudocode 2.2 Pseudocode penerapan prim trajectory pada minimum spanning tree



Gambar 2.4. Contoh graph yang telah dilakukan Prim Trajectory

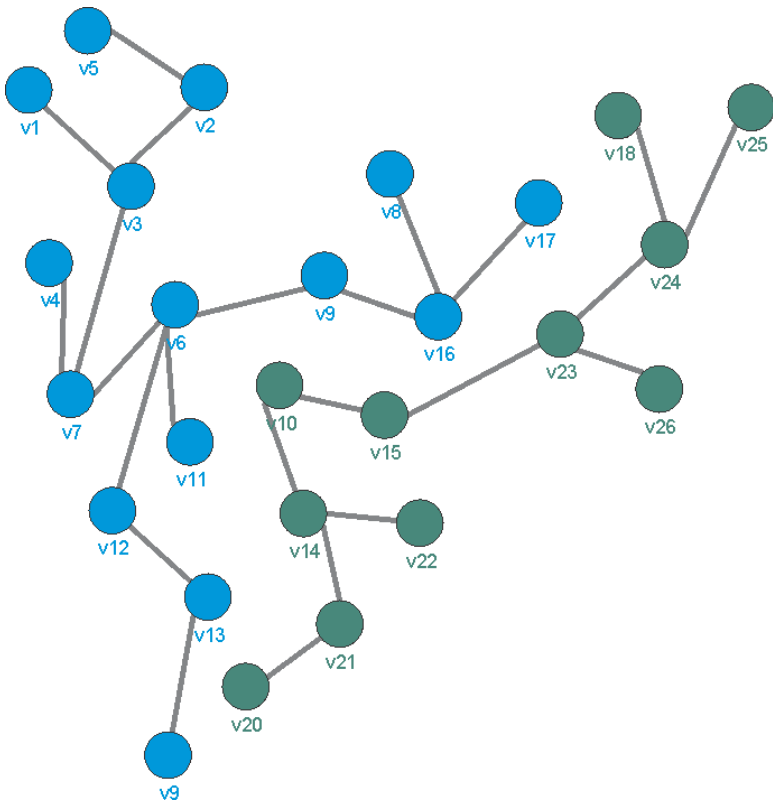
C. Penghitungan Jarak Cluster ke Vertex Terdekat

Pada Proses Clustering K-Means vertex yang dihapus dari hasil praproses data Prim Trajectory akan dikelompokkan kedalam kelas yang paling dekat. Sedangkan data hasil Prim Trajectory / subset yang terbentuk akan digunakan dan dianggap sebagai sebuah centroid awal pada proses Clustering K-Means. Selanjutnya sisa data yang belum tergabung ke dalam subset akan dihitung jarak terdekat ke sebuah subset dengan menggunakan rumus dan dimasukkan kedalam subset terdekat sesuai dengan Pseudocode 2.3 [6] dan menghasilkan Gambar 2.5.

Input	: initial cluster (C_i) , Peneliti yang belum memiliki kelas (outer), minimum spanning tree yang dihasilkan pada proses awal MST.
Output	: graph X akhir yang telah dicluster
Pseudocode	:

1. Ulang untuk semua outer O
2. indeksV = vertex C_i yang memiliki jarak terpendek dengan O
3. Class = indeks dari cluster vertex indeksV
4. C_{class}.append(O)

Pseudocode 2.3 Pseudocode penghitungan jarak cluster ke vertex terdekat

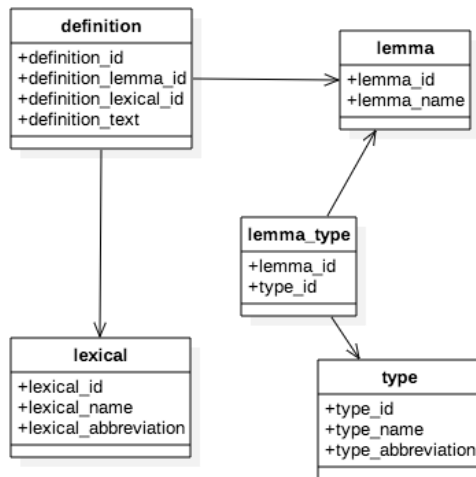


Gambar 2.5. Hasil Clustering

2.3 Kamus Bahasa Kateglo Dataset

Kateglo merupakan website yang menyediakan kamus, tesaurus dan glosarium bahasa Indonesia online. Website tersebut berisikan kamus, tesaurus, dan glosarium (daftar istilah) untuk kata-kata dalam bahasa Indonesia. Saat ini sudah ada sekitar 192.000 entri glosarium dan lebih dari 70.000 entri kamus. Selain itu, kateglo juga menyediakan informasi tentang peribahasa (proverb) yang sampai saat ini memiliki koleksi lebih dari 2000 peribahasa, dan juga menyediakan daftar akronim dan singkatan yang sudah terinput sekitar 3400 entri. Namun sayang belum ada keterangan berapa entri untuk daftar tesaurusnya.

Dalam menyusun thesaurus dalam tugas akhir ini memanfaatkan dataset yang disediakan kateglo. Gambar 2.6 dan Tabel 2.4 merupakan PDM dataset kateglo yang digunakan dalam tugas akhir ini :



Gambar 2.6. PDM Database Kateglo

Tabel 2.4. Tabel Detail Database Kateglo

No	Tabel	Keterangan	Jumlah record
1	Definition	Record merupakan definisi dari lemma. 1 lemma bisa memiliki lebih dari 1 definisi	100.066
2	Lemma	Record merupakan lemma atau kata dalam bahasa indonesia	225.720
3	Lexical	Record merupakan tipe dari setiap definisi yang mewakili lemma (noun, verb, adjective, dll)	7
4	Type	Record merupakan tipe dari lemma (kata dasar, imbuhan, kata turunan, peribahasa)	4
5	Lemma_type	Record merupakan hasil pivot dari tabel lemma dan tabel type	72.427

Setiap kata yang terdapat pada Kateglo memiliki leksikal, yaitu kelas kata yang digunakan untuk penggolongan kata dalam satuan bahasa berdasarkan kategori bentuk, fungsi, dan makna dalam sistem gramatikal. Contoh dari kata dan leksikal kata dijelaskan sebagai berikut:

A. Adjektiva

Adjektiva (kata sifat) adalah kata yang digunakan untuk memberi sifat, menambah suatu makna pada suatu kata benda atau kata ganti.

B. Adverbia

Adverbia (kata keterangan) adalah kata yang digunakan untuk memberikan informasi lebih banyak tentang kata kerja, kata keterangan yang lain, atau keseluruhan kalimat.

C. Nomina

Nomina (kata benda) adalah kata atau kelompok kata yang menyatakan suatu nama.

D. Numeralia

Numeralia adalah kata atau frasa yang menunjukkan bilangan atau kuantitas.

E. Pronomina

Pronomina (kata ganti) adalah kata yang digunakan sebagai kata benda atau frase kata benda.

F. Verba

Verba (kata kerja) adalah kata yang digunakan untuk menyatakan suatu perbuatan, kejadian, peristiwa, eksistensi, pengalaman, keadaan antara dua benda. Tabel 2.5 merupakan contoh kata berdasarkan leksikal kata

Tabel 2.5 Tabel Contoh Kata Berdasarkan Tipe Leksikal

No	Tipe leksikal	Contoh kata
1	Adjektiva	keras, jauh, dan kaya.
2	Adverbia	cuman, amat, sangat dan konon.
3	Nomina	mesin, mikrofon, minyak dan minuman.
4	Numeralia	berenam, bertiga, delapan dan miliar.
5	Pronomina	kau, kaulah, kamu dan saya.
6	Verba	antre, alur, gagal dan goreng.

2.4 Cluster Analisis (*Dunn Index*)

Dunn index (DI) (diperkenalkan oleh J. C. Dunn pada 1974) adalah sebuah perhitungan untuk menentukan evaluasi suatu

cluster. Tujuan dari Dunn Index ini adalah untuk mendapat nilai kedekatan suatu cluster dengan variance antar anggota cluster yang kecil. Semakin besar nilai dari Dunn Index semakin bagus hasil dari metode clustering yang dipakai [8]. Dunn Index akan dipakai untuk menentukan nilai K yang paling optimal yang dihasilkan pada K-Means clustering berbasis graph. Berikut cara menghitung Dunn Index sesuai dengan Persamaan 2.4 dan Persamaan 2.5 :

$$\delta(C_i) = \frac{1}{N_{C_i}} \sum_{j=1}^{N_{C_i}} x_{i,j}$$

Dimana :

- i = iterasi cluster, $0 < i \leq \text{jumlah cluster}$
- C_i = Cluster dengan index i
- $\delta(C_i)$ = *Intercluster Distance* pada C_i
- N_{C_i} = Jumlah edge pada C_i
- $x_{i,j}$ = nilai edge dari C_i dengan indeks j

Persamaan 2.4. Rumus Perhitungan Intercluster Distance dari suatu graph

$$DI = \frac{\min_{1 \leq i \leq m} (\delta(C_i))}{\max(edge)}$$

Dimana :

- m = jumlah cluster
- i = iterasi cluster, $0 < i \leq m$
- DI = *Dunn Index*
- $edge$ = nilai edge dari seluruh graph

Persamaan 2.5. Rumus Perhitungan Dunn Index dari suatu graph

2.5 Betweenness Centrality

Betweenness centrality atau disebut juga nilai tengah adalah sebuah indikator sentralitas sebuah vertex dalam jaringan graph. Betweenness centrality sama dengan angka dari jalur terpendek dari semua vertex ke semua vertex yang melewati vertex tersebut. Sebuah vertex dengan nilai betweenness centrality yang tinggi mempunyai pengaruh yang besar dalam suatu graph, dengan asumsi bahwa jalur yang dilewati adalah jalur terpendek.

Dalam tugas akhir ini metode betweenness centrality digunakan untuk mencari vertex peneliti mana memiliki nilai betweenness centrality tertinggi sehingga memiliki pengaruh dalam riset yang ada di ITS. Berikut rumus perhitungan betweenness centrality sesuai dengan Persamaan 2.6 [9].

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Dimana :

- v = vertex dalam graph
- $g(v)$ = nilai betweenness centrality dari vertex
- s = vertex asal
- t = vertex tujuan
- σ_{st} = jumlah banyaknya jarak terpendek dari vertex s ke vertex t
- $\sigma_{st}(v)$ = jumlah banyaknya jarak terpendek dari vertex s ke vertex t yang melewati vertex v

Persamaan 2.6 Rumus menghitung betweenness centrality dari suatu vertex

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas mengenai perancangan dan analisis tugas akhir. Tugas akhir ini akan memproses clustering peneliti berdasarkan topik peneliti menggunakan *k-means berbasis graph* dan *expand k-means berbasis graph dengan thesaurus yang dibentuk dari database kataglo*.

3.1 Analisis

Analisis sistem terbagi menjadi dua bagian, yaitu analisis permasalahan yang diangkat pada tugas akhir dan gambaran umum sistem yang dibangun.

3.1.1 Analisis Permasalahan

Permasalahan yang diangkat dari tugas akhir ini adalah bagaimana mengetahui kelompok-kelompok fokus penelitian yang ada di ITS. Pengelompokan dapat dilakukan secara manual dengan melakukan pengecekan pada database Pusat Data ITS namun membutuhkan waktu yang lama untuk mendapatkan informasi tersebut. Saat ini visualisasi kerjasama peneliti ITS dalam bentuk sudah tersedia namun masih belum dapat ditarik informasi mengenai kelompok-kelompok penelitian yang terbentuk dari fokus topik penelitian yang ada di ITS.

Permasalahan diatas dapat di selesaikan dengan menerapkan metode graph clustering pada graph kerjasama peneliti ITS. Dari cluster yang didapat kita mendapatkan fokus topik tiap cluster. Metode clustering yang dipakai dalam tugas akhir ini adalah K-Means berbasis graph.

Untuk menambah akurasi dalam clustering pada tugas akhir ini juga dikembangkan pembentukan graph kerjasama peneliti ITS dengan memperhatikan sinonim topik antar peneliti. Diharapkan penghitungan bobot topik antar peneliti dengan sinonim kata menghasilkan cluster yang lebih baik. Dalam

penghitungan bobot nantinya akan beberapa rumus dan algoritma yang di modifikasi khususnya saat pembentukan graph peneliti.

3.1.2 Deskripsi Umum Sistem

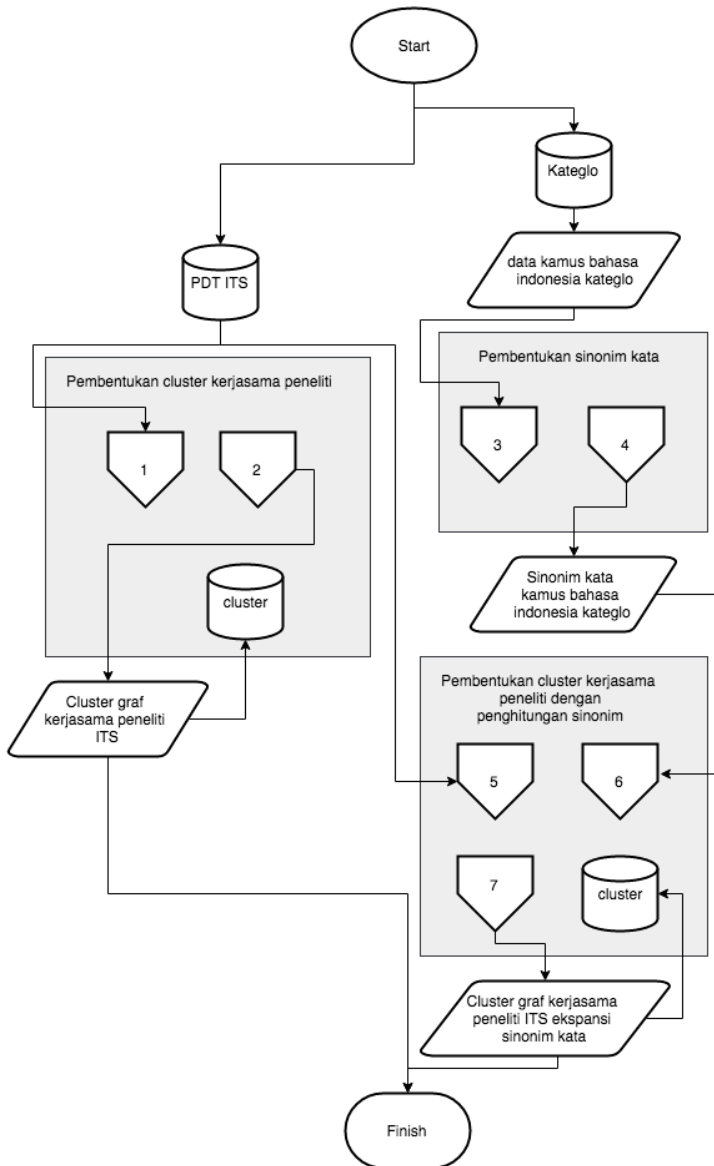
Sistem yang dibangun pada tugas akhir ini akan dibagi menjadi 3 proses utama, yaitu clusterisasi data kerjasama peneliti ITS dengan metode K-Means berbasis graph, pembentukan sinonim kata dengan menggunakan database kataglo, expand K-Means berbasis graph dengan sinonim kata. Berikut alur sistem pada tugas akhir ini.

3.2 Perancangan

Perancangan tugas akhir ini dibagi menjadi empat bagian bahasan yaitu bahasa pemrograman python yang akan digunakan sebagai bahasa pemrograman yang utama dalam tugas akhir ini, Clustering K-means berbasis graf, pembentukan sinonim thesaurus dari database kataglo, dan Ekspansi pembentukan graph kerjasama peneliti ITS dengan sinonim thesaurus yang terbentuk dari proses sebelumnya. Gambaran umum alur sistem pada tugas akhir ini dapat dilihat pada Gambar 3.1.

3.2.1 Bahasa Pemrograman Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Dalam pengerjaan tugas akhir ini digunakan beberapa pustaka python diantaranya.



Gambar 3.1 Deskripsi Umum Sistem [Main]

A. NLTK Tokenizer

Natural Language Processing (NLP) merupakan salah satu cabang ilmu yang berfokus pada pengolahan bahasa alami manusia (bahasa inggris, jerman, Indonesia dan lain-lain) yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Dalam Natural Language Processing (NLP) banyak bahasa pemrograman dan tools yang bisa digunakan. NLTK merupakan modul dalam bahasa python yang dapat digunakan untuk melakukan tokenisasi kalimat pada saat tahap praproses data.

B. Sastrawi

Pustaka sastrawi dalam python adalah satu-satunya stemmer bahasa indonesia yang tersedia dalam python. Pada awal pembentukan pustaka sastrawi dataset yang digunakan juga berasal dari kateglo.

C. PYMSSQL

Pymssql adalah suatu pustaka dalam bahasa python untuk melakukan koneksi dengan database Microsoft SQL Server. Pada tugas akhir ini pymssql digunakan untuk melakukan koneksi ke database repositori peneliti Institut Teknologi Sepuluh Nopember.

D. MYSQL Connector

MYSQL Connector merupakan pustaka dalam bahasa pemrograman python untuk melakukan koneksi ke database mysql. Pada tugas akhir ini pustaka MYSQL Connector digunakan untuk melakukan koneksi ke database kateglo untuk menyusun sinonim thesaurus.

3.2.2 Pembentukan Cluster Kerjasama Peneliti

Dalam melakukan clustering pada data kerjasama peneliti ITS ada beberapa metode yang harus disesuaikan dari algoritma k-means berbasis graph seperti pada penjelasan bab sebelumnya terhadap data yang dimiliki oleh Sistem Repositori Peneliti ITS. Alur dari pembentukan cluster kerjasama peneliti bisa dilihat pada Gambar 3.2.

Rancangan clustering data peneliti menggunakan k-means berbasis graph dimulai dengan mengubah bentuk graph data peneliti ITS dari nilai kerjasama kedalam nilai edge. Karena semakin besar nilai kerjasama berarti semakin besar kedekatan antar dua peneliti yang bertolak belakang dengan konsep nilai edge yaitu semakin kecil nilai edge maka semakin dekat jarak antar vertexnya. Setelah itu nilai edge dan vertex yang berupa undirected graph diproses dengan algoritma prim membentuk minimum spanning tree dari graph kerjasama peneliti ITS. Graph minimum spanning tree yang terbentuk lalu dilakukan Thresholding untuk mendapatkan subset graph yang nantinya akan menjadi initial cluster dari K-means. Tahap akhir dari metode ini adalah pengelompokan vertex yang belum memiliki cluster kedalam initial cluster yang terbentuk.

A. Inisiasi Edge Value

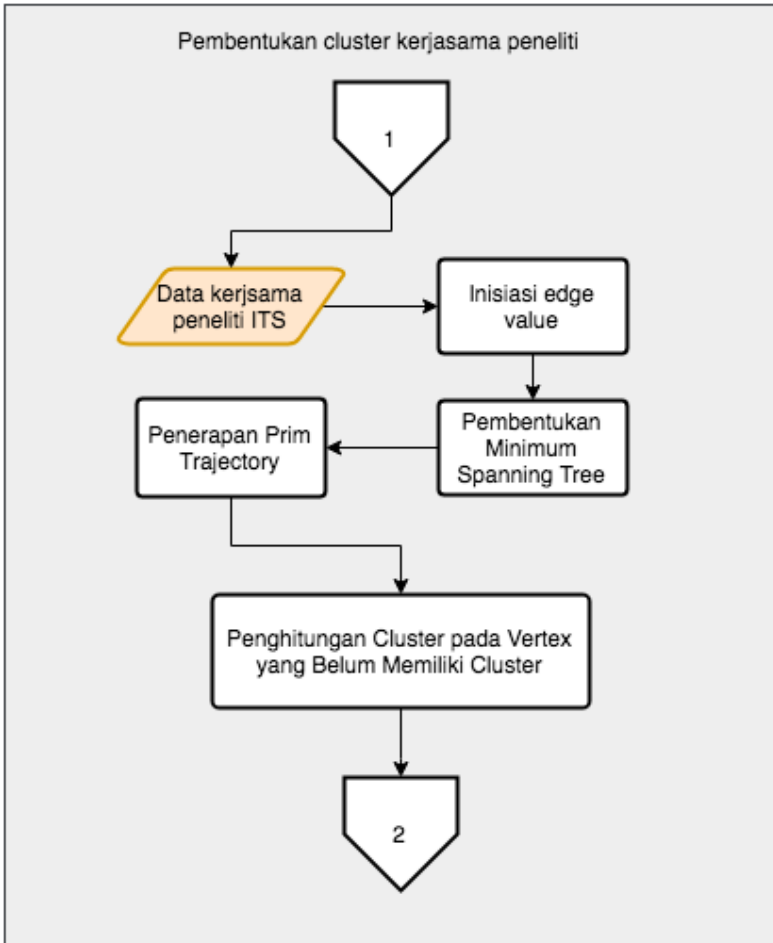
Pada data repositori peneliti ITS besarnya nilai kerjasama berbanding lurus dengan kedekatan topik kerjasama sedangkan prinsip nilai edge yang dipakai dalam metode k-means berbasis graph adalah semakin kecil nilai edge maka semakin besar kedekatan antar vertex. Oleh karena itu perlu untuk dilakukan pengolahan data terlebih dahulu dengan rumus :

$$Edge_{i=1}^N = (\max(k) + 0.00001) - k_i$$

dimana :

N = jumlah banyaknya semua edge

k = nilai kerjasama antar peneliti
 k_i = nilai kerjasama ke i
 $\max(k)$ = nilai maksimal dari k



Gambar 3.2. Tahapan Proses Pembentukan Cluster Kerjasama Peneliti

Setelah menerapkan rumus diatas maka nilai edge sudah sesuai dengan k-means berbasis graph. Misalkan terdapat nilai bobot kerjasama peneliti seperti tabel Tabel 3.1.

Tabel 3.1 Data Kerjasama Peneliti ITS

id	kode_peneliti1	kode_peneliti2	bobot_kerjasama
1	569	542	0.36
2	654	236	0.65
3	236	245	0.99
...
67	365	458	1.36
...
N	125	156	0.12

Langkah-langkah yang dilakukan untuk mendapatkan nilai jarak edge dari data kerjasama peneliti ITS adalah sebagai berikut:

- Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- Pencarian data dengan query ke basis data semua nilai bobot kerjasama dosen peneliti ITS.
- Pencarian data dengan query ke basis data mendapatkan nilai bobot kerjasama maksimal dari semua nilai bobot kerjasama.
- Ulangi untuk semua data bobot kerjasama dosen peneliti ITS dan perbarui nilai bobot dengan rumus yang tercantum diatas yaitu, nilai maksimal edge + 0.00001 – nilai bobot.
- Simpan semua data edge ke dalam basis data hasil pengolahan nilai bobot kerjasama ke dalam nilai jarak edge.

Sehingga apabila Tabel 3.2-1 di proses dengan langkah-langkah diatas, dan dimisalkan data dengan id sama dengan 67 mempunyai nilai bobot kerjasama maksimal maka akan didapat hasil akhir seperti pada Tabel 3.2.

Tabel 3.2 Data Kerjasama Peneliti ITS dalam bentuk nilai jarak edge

id	kode_peneliti1	kode_peneliti2	Nilai jarak edge
1	569	542	1.00001
2	654	236	0.71001
3	236	245	0.37001
...
67	365	458	0.00001
...
N	125	156	1.24001

B. Pembentukan Minimum Spanning Tree

Ada beberapa macam algoritma untuk membentuk minimum spanning tree seperti prim, kruskal, dan lain lain. Pada tugas akhir ini algoritma yang dipakai untuk membentuk minimum spanning tree adalah algoritma prim. Algoritma memiliki sifat brute force (mencoba semua kemungkinan) pada edge hingga semua vertex diproses dan memiliki nilai edge minimal. Berikut algoritma dari prim minimum spanning tree :

- Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- Inisiasi variabel flag array dengan indeks kode peneliti dengan nilai boolean 1 atau 0 untuk menyimpan penanda apakah peneliti sudah pernah dikunjungi atau belum dalam membuat tree.
- Inisiasi array variabel mst untuk menampung minimum spanning tree yang akan terbentuk dengan nilai terdiri dari tiga nilai struct yaitu (vertex asal, vertex tujuan, nilai edge).
- Query vertex pada basis data secara acak untuk dijadikan initial vertex pada mst.

- e) Query semua edge yang terhubung dengan vertex initial, urutkan hasil query berdasarkan nilai edge dari terkecil ke terbesar lalu simpan semua hasil query dalam sebuah variabel berstruktur queue.
- f) Tandai peneliti initial vertex pada variabel flag telah dikunjungi
- g) Lakukan perulangan hingga queue kosong:
- h) Inisiasi variabel dengan tipe struct tmpVertex = nilai queue dengan index 0 atau bisa disebut dengan queue.front()
- i) Jika flag vertex asal dan vertex tujuan dari tmpVertex salah satunya ada yang belum dikunjungi, maka query ke basis data semua edge yang memiliki hubungan dengan vertex asal atau vertex tujuan yang belum dikunjungi dan tambahkan hasil query ke dalam queue
- j) Tandai vertex asal dan vertex tujuan pada tmpVertex sudah dikunjungi.
- k) Urutkan kembali queue dengan urutan menaik sesuai dengan nilai dari edge
- l) Keluarkan queue dengan index 0 dari queue atau biasa disebut dengan fungsi queue.pop()
- m) Cek kondisi pada langkah (g) jika masih terpenuhi maka kembali ke langkah (g), jika sudah tidak memenuhi maka mst sudah terbentuk.

Setelah semua langkah diatas diterapkan maka akan didapat minimum spanning tree dari data kerjasama peneliti. Seperti contoh berikut data sebelum (Gambar 3.3) dan sesudah (Gambar 3.4) diterapkan metode prim untuk mendapatkan minimum spanning tree.



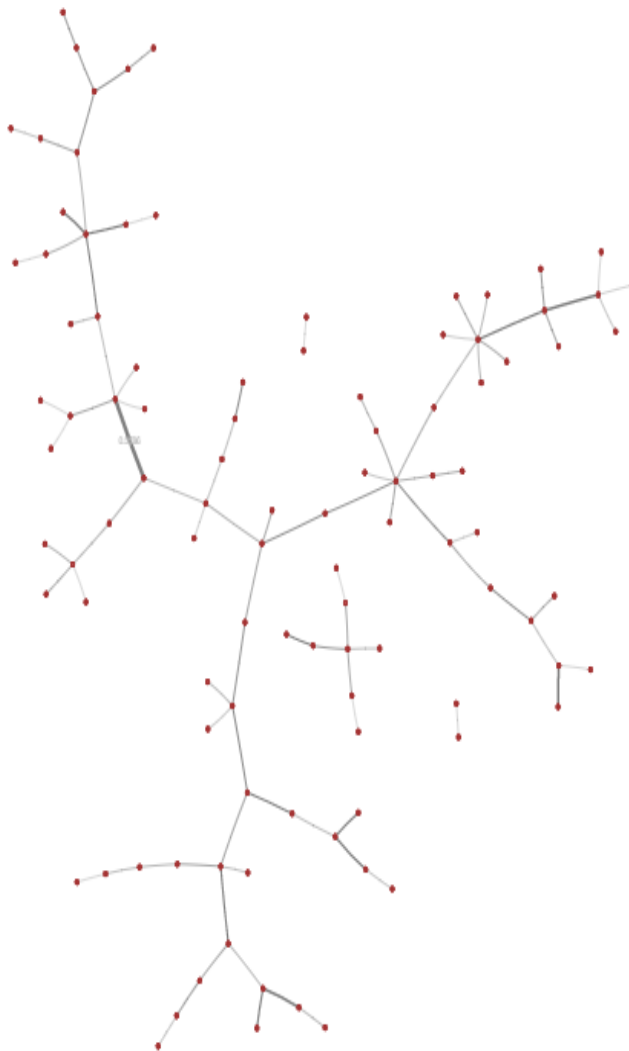
Gambar 3.3 Graph kerjasama peneliti pada Fakultas Teknologi Informasi ITS sebelum di terapkan metode mst

C. Penerapan Prim Trajectory

Prim Trajectory merupakan suatu metode mendapatkan nilai K pada k-means berbasis graph dan untuk menentukan initial

cluster untuk digunakan pada langkah selanjutnya. Prim Trajectory memproses minimum spanning tree yang dihasilkan pada langkah sebelumnya untuk menjadi initial cluster. Prim trajectory menggunakan konstanta Threshold sebagai indikator sebuah subset graph dapat dikatakan sebagai initial cluster atau bukan. Subset graph yang memenuhi nilai Threshold akan dijadikan initial cluster. Berikut algoritma dari prim trajectory :

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi variabel flag array dengan indeks kode peneliti dengan nilai boolean 1 atau 0 untuk menyimpan penanda apakah peneliti sudah pernah dikunjungi atau belum dalam membuat mst.
- c) Inisiasi array variabel adjacency matriks mst dan isi variabel mst dengan graph mst yang terbentuk pada proses sebelumnya dengan nilai edge kurang dari konstanta C.
- d) Inisiasi array variabel adjacency matriks initial cluster dengan tipe struct sebagai berikut (vertex asal, vertex tujuan, nilai edge, kelas).
- e) Ulangi untuk semua mst dengan iterator variabel v:
- f) Jika flag peneliti pada vertex asal v belum dikunjungi maka traverse graph variabel mst dan hitung semua vertex yang dimiliki oleh subgraph yang terbentuk dari traverse graph tersebut.
- g) Jika jumlah vertex hasil traverse lebih besar dari nilai konstanta N maka tambahkan semua edge hasil traverse graph ke dalam initial cluster dan tandai flag semua vertex hasil traverse menjadi telah dikunjungi.
- h) Jika perulangan pada langkah (e) selesai maka initial cluster telah terbentuk.



Gambar 3.4 Minimum Spanning Tree yang terbentuk dari data graph kerjasama peneliti Fakultas Teknologi Informasi ITS

Nilai C dan N dapat ditentukan sesuai kebutuhan saat ujicoba pada data yang akan diproses. Nilai N dan C akan sangat berpengaruh pada jumlah initial cluster yang akan dihasilkan.

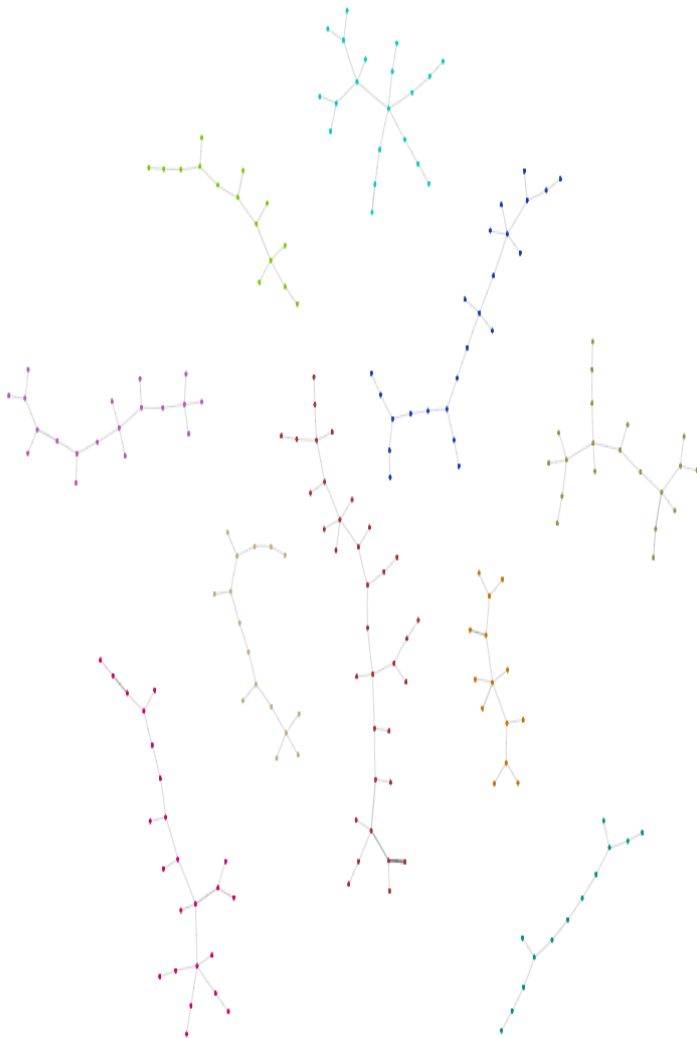
Traverse graph merupakan sebuah metode menjelajah graph yang masih terhubung. Penelusuran ini perlu dilakukan karena hasil thresholding dari konstanta C akan membuat mst terpecah-pecah menjadi subgraph. Subgraph inilah yang akan menjadi calon initial cluster, namun thresholding selanjutnya dengan konstanta N maka akan mendapat batas minimal jumlah minimal vertex suatu subgraph bisa dimasukkan kedalam kriteria initial cluster sesuai dengan Gambar 3.5.

D. Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster

Setelah mendapatkan initial cluster dari langkah prim trajectory ada node yang masih belum memiliki kelas. Pada langkah ini node-node tersebut akan di masukkan ke dalam initial cluster yang terbentuk.

Algoritma pada langkah ini cukup sederhana yaitu dengan mencari node terdekat pada minimum spanning tree yang telah tergabung ke dalam initial cluster.

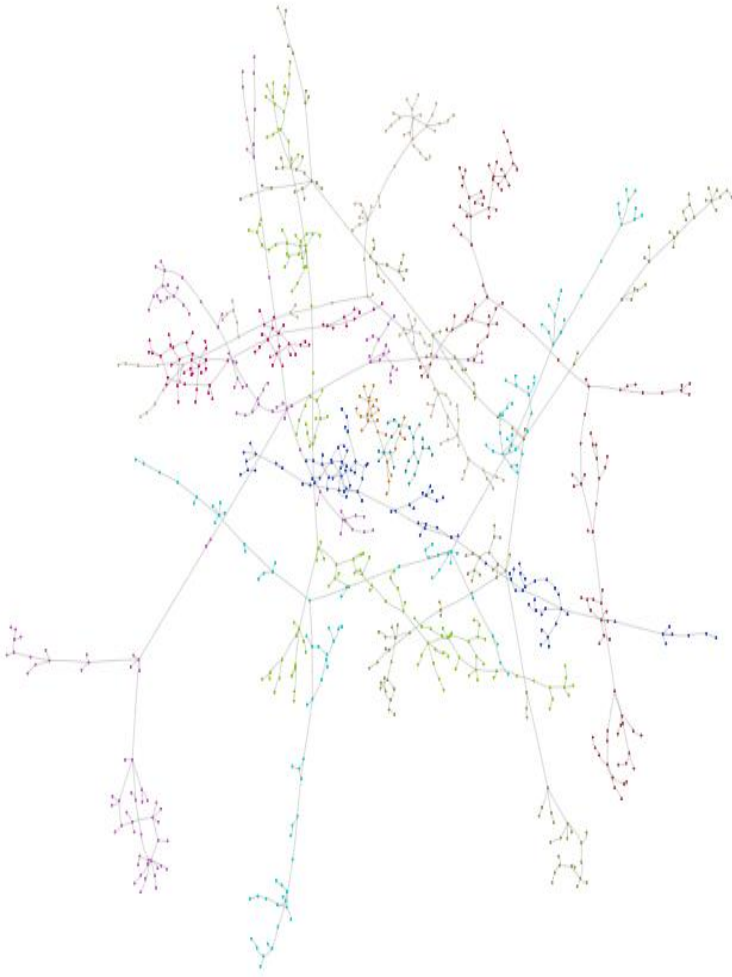
- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array variabel adjacency matriks incluster dan query dari basis data semua edge yang terdapat pada initial cluster hasil dari pengolahan prim trajectory diatas
- c) Inisiasi array variabel adjacency matriks outcluster dan isi dengan query dari basis data semua edge yang belum termasuk dalam initial cluster.



Gambar 3.5 Initial cluster yang terbentuk dari hasil pengolahan minimum spanning tree dengan prim trajectory

- d) Inisiasi array variabel adjacency matriks initial cluster dengan tipe struct sebagai berikut (vertex asal, vertex tujuan, nilai edge, kelas).
- e) Inisiasi variabel flag array dengan indeks kode peneliti dan set flag sudah dikunjungi untuk semua vertex yang terdapat pada array incluster.
- f) Inisiasi variabel queue dan isi dengan semua edge yang terhubung dengan graph initial cluster. Sehingga queue berisi semua edge tepi dari initial cluster. Edge tepi adalah edge yang hanya memiliki 1 tetangga edge saja dari 2 vertex yang dimiliki.
- g) Urutkan queue dengan urutan menaik berdasarkan nilai dari edge.
- h) Lakukan perulangan hingga queue kosong:
- i) Inisiasi variabel dengan tipe struct `tmpVertex` = nilai queue dengan index 0 atau bisa disebut dengan `queue.front()`
- j) Jika flag vertex asal dan vertex tujuan dari `tmpVertex` salah satunya ada yang belum dikunjungi, maka query ke basis data semua edge yang memiliki hubungan dengan vertex asal atau vertex tujuan yang belum dikunjungi dan tambahkan hasil query ke dalam queue
- k) Tandai vertex asal dan vertex tujuan pada `tmpVertex` sudah dikunjungi.
- l) Urutkan kembali queue dengan urutan menaik sesuai dengan nilai dari edge
- m) Keluarkan queue dengan index 0 dari queue atau biasa disebut dengan fungsi `queue.pop()`
- n) Cek kondisi pada langkah (h) jika masih terpenuhi maka kembali ke langkah (h), jika sudah tidak memenuhi maka proses sudah selesai.

Dengan algoritma diatas semua node yang belum memiliki kelas akan dimasukkan kedalam kelas terdekat sesuai dengan Gambar 3.6.



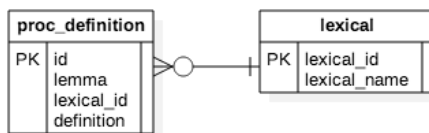
Gambar 3.6 Graph Kerjasama Setelah Diterapkan Metode K-Means berbasis Graph

3.2.3 Pembentukan Sinonim Thesaurus Kateglo

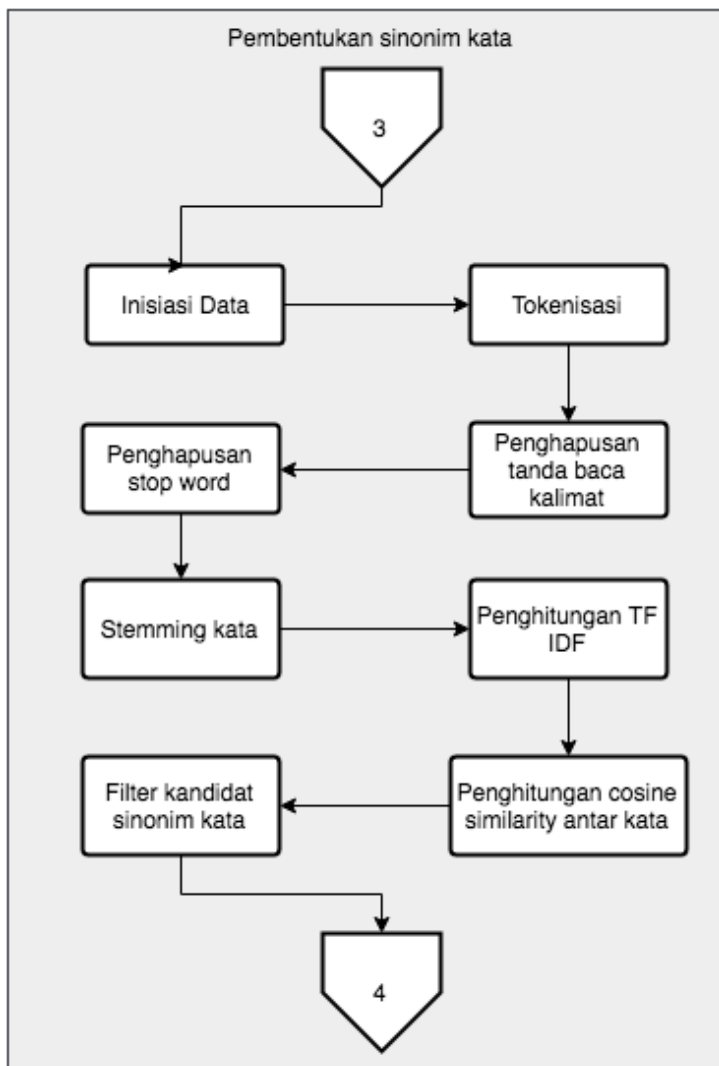
Dalam pembentukan sinonim thesaurus banyak metode yang bisa dipakai untuk mendapatkan hubungan keterkaitan antar-kata. Pada umumnya kata yang bersinonim memiliki pengertian atau definisi yang memiliki kemiripan tinggi, sehingga dengan memiliki arti dari setiap kata kita dapat mendapatkan sinonim dari kata dengan mencari kemiripan antar kata. Definisi kata dapat kita dapatkan dari database kateglo. Kateglo merupakan glosarium bahasa indonesia online. Dengan algoritma pemrosesan teks akan didapat kemiripan definisi antar kata. Gambar 3.8 merupakan alur pembentukan sinonim thesaurus dari database kateglo.

A. Filter kata yang mempunyai sifat kata kerja dan kata benda

Tentunya dalam pembentukan thesaurus ini dibutuhkan efisiensi waktu eksekusi sehingga perlu membuat database kateglo menjadi lebih simpel dan menghilangkan fitur-fitur yang tidak dipakai. seperti contoh kata yang dibutuhkan hanya berupa kata dasar, karena kata yang diolah nanti akan dilakukan stemming yang merupakan mengubah kata menjadi kata dasar. Gambar 3.7 dan Tabel 3.3 merupakan struktur tabel yang akan digunakan untuk pembentukan thesaurus dari database kateglo.



Gambar 3.7. Struktur tabel pembentukan thesaurus



Gambar 3.8 Alur pembentukan sinonim dari database katego

Tabel 3.3. Keterangan database pembentukan thesaurus

Kolom	Tabel	Keterangan
id	proc_definition	Primary key dari tabel proc_definition
lemma	proc_definition	Berisi kata dalam bahasa indonesia
lexical_id	proc_definition	Foreign_key dari tabel lexical
definition	proc_definition	Definisi dari kolom lemma
lexical_id	lexical	Primary key dari tabel lexical
lexical_name	lexical	Tipe dari suatu kata (noun, verb, adverb, dll)

Pada proses selanjutnya setiap record dari tabel proc_definition akan dianggap sebagai dokumen dan kata adalah berasal dari kolom definition.

Contoh kasus yang akan dibahas dalam bab ini adalah lemma “warta” dengan definisi “berita; kabar” dan lemma “informasi” dengan definisi “pemberitahuan; kabar atau berita tentang sesuatu”.

B. Tokenisasi

Proses tokenisasi dilakukan pada setiap definisi dari setiap kata yang ada. Tokenisasi adalah suatu proses untuk membagi suatu teks berupa kalimat atau paragraf menjadi unit-unit kecil berupa kumpulan kata atau token [4].

Sebelum melakukan tokenisasi, kalimat definisi setiap lemma disimpan pada suatu variabel. Tokenisasi dilakukan dengan menggunakan fungsi pustaka pada bahasa pemrograman python , yaitu ***word_tokenize()*** pada pustaka ***nlk***. Fungsi ***word_tokenize()*** digunakan untuk memisahkan kata pada suatu kalimat berdasarkan spasi. Sehingga lemma “warta” dengan definisi “berita; kabar” dan lemma “informasi” dengan definisi “pemberitahuan; kabar atau

berita tentang sesuatu” apabila diterapkan metode tokenisasi menjadi token sebagai berikut:

$$Token_{warta} = \begin{bmatrix} "berita;" \\ "kabar" \end{bmatrix}$$

$$Token_{informasi} = \begin{bmatrix} "pemberitahuan;" \\ "kabar" \\ "atau" \\ "berita" \\ "tentang" \\ "sesuatu" \end{bmatrix}$$

Kata-kata yang dihasilkan dari proses tokenisasi disimpan dalam variabel berbentuk *array* yang selanjutnya akan digunakan pada proses penghapusan tanda baca.

C. Penghapusan Tanda Baca Kalimat

Proses penghapusan tanda baca ini dilakukan pada data definisi yang telah di lakukan tokenisasi. Penghapusan tanda baca dilakukan karena terkadang hasil dari tokenisasi masih menyisakan tanda baca dalam token kata yang terbentuk. Berikut contoh hasil dari penghapusan tanda baca pada token kata yang dihasilkan dari tokenisasi:

$$Token_{warta} = \begin{bmatrix} "berita" \\ "kabar" \end{bmatrix}$$

$$Token_{informasi} = \begin{bmatrix} "pemberitahuan" \\ "kabar" \\ "atau" \\ "berita" \\ "tentang" \\ "sesuatu" \end{bmatrix}$$

Penghapusan tanda baca pada tugas akhir ini menggunakan bahasa pemrograman python dengan pustaka *string* dan fungsi *string.punctuation*. setelah semua token sudah dihilangkan semua tanda bacanya maka setiap token siap untuk dicek termasuk kedalam stop word atau tidak.

D. Penghapusan Stop Word

Proses selanjutnya adalah melakukan penghapusan *stopword* yang terdapat pada kumpulan kata-kata hasil tokenisasi dan penghapusan tanda baca. Terdapat suatu berkas kumpulan *stopword* yang sering muncul pada suatu artikel atau dokumen. Berkas kumpulan *stopword* didapatkan dari penyedia list *stopword* bahasa Indonesia.

Kemudian, yang dilakukan pada isi dari berkas kumpulan *stopword* tersebut adalah membandingkannya dengan variabel *array* yang menyimpan hasil tokenisasi. Jika kata yang tersimpan pada variabel *array* terdapat pada berkas kumpulan *stopword*, kata tersebut dihapus dari variabel *array* sehingga pada proses pembentukan sinonim penghitungan bobot kata, kata tersebut tidak akan menjadi kata terdaftar. Sebagai contoh dalam kedua token kata diatas kata-kata yang masuk kedalam list stop word ada 3 yaitu kata “atau”, “tentang”, dan “sesuatu”. Sehingga token kata menjadi:

$$Token_{warta} = \begin{bmatrix} "berita" \\ "kabar" \end{bmatrix}$$

$$Token_{informasi} = \begin{bmatrix} "pemberitahuan" \\ "kabar" \\ "berita" \end{bmatrix}$$

Tujuan dari proses ini adalah agar kata yang digunakan pada proses *stemming* dan pembentukan sinonim penghitungan bobot kata merupakan kata-kata yang memiliki informasi penting, bukan kata-kata yang sering muncul pada artikel definisi suatu lemma.

E. Stemming Kata

Stemming adalah proses untuk mengembalikan bentuk dari suatu kata pada bentuk dasar kata tersebut [4]. Setelah *stopword* dihapus dari variabel *array*, dilakukan proses *stemming* terhadap kata-kata yang masih tersimpan di dalamnya. Sehingga apabila contoh kasus diatas diterapkan metode stemming menjadi :

$$Token_{warta} = \begin{bmatrix} "berita" \\ "kabar" \end{bmatrix}$$

$$Token_{informasi} = \begin{bmatrix} "tahu" \\ "kabar" \\ "berita" \end{bmatrix}$$

Proses *stemming* dilakukan dengan memanfaatkan pustaka ***Sastrawi.Stemmer.StemmerFactory.create_stemmer*** dengan fungsi ***stem()***. Pustaka *sastrawi* merupakan pustaka stemmer bahasa indonesia dalam bahasa python. Setelah seluruh kata pada variabel *array* melalui proses *stemming*, simpan kata-kata tersebut ke dalam basis data untuk digunakan pada proses penghitungan TF x IDF.

F. Penghitungan TF x IDF

Setiap token yang sudah dibersihkan list stop wordnya dan sudah dilakukan stemming lalu dihitung TF nya terlebih dahulu. Berikut merupakan algoritma untuk menghitung TF :

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, *pymssql*.
- b) Inisiasi array lemma dengan index lemma dan isi dengan semua token yang telah di stemming.
- c) Ulangi untuk semua item dalam variabel lemma:
- d) Ulangi untuk semua token yang terdapat pada item:

- e) $F_{\text{item},\text{token}}++$
- f) Jika semua token pada perulangan (d) sudah diiterasi lanjutkan ke (g) jika tidak kembali ke (d)
- g) Ulangi untuk semua nilai yang terdapat pada variabel F:
- h) $TF_{\text{item},\text{token}} = 1 + \log_{10} F_{\text{item},\text{token}}$
- i) Jika semua nilai F pada perulangan (g) sudah diiterasi lanjutkan ke (j) jika tidak kembali ke (g)
- j) Jika semua item pada perulangan (c) sudah diiterasi lanjutkan ke (k) jika tidak kembali ke (c)
- k) Simpan hasil TF ke dalam basis data.

Setelah langkah-langkah diatas diterapkan pada token contoh seperti dilangkah stemming, maka didapat nilai TF

$$TF_{\text{ warta}} = \begin{bmatrix} \text{berita}, 1 \\ \text{kabar}, 1 \end{bmatrix}$$

$$TF_{\text{informasi}} = \begin{bmatrix} \text{tahu}, 1 \\ \text{kabar}, 1 \\ \text{berita}, 1 \end{bmatrix}$$

Setelah itu dilakukan penghitungan IDF dengan algoritma sebagai berikut :

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array term dengan index term dan isi dengan nilai query ke basis data banyaknya term muncul dalam semua dokumen.
- c) Ulangi untuk semua item dalam variabel lemma:
- d) $IDF_{\text{item}} = \log_{10} (\text{Jumlah dokumen} + 1) / \text{lemma}[\text{item}]$
- e) Jika semua item pada perulangan (c) sudah diiterasi lanjutkan ke (f) jika tidak kembali ke (c)
- f) Simpan hasil IDF ke dalam basis data.

Setelah langkah-langkah penghitungan IDF diatas selesai maka untuk contoh kasus didapat nilai IDF sebagai berikut:

$$IDF = \begin{bmatrix} tahu, 0.47712 \\ kabar, 0.17609 \\ berita, 0.17609 \end{bmatrix}$$

Tahap akhir dari tahapan ini adalah dengan mengalikan hasil tiap TF x IDF dengan langkah-langkah sebagai berikut :

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array idf dengan index term kata dan isi dengan query dari basis data dengan nilai idf dari masing-masing term.
- c) Inisiasi array lemma dengan index lemma dan isi dengan query dari basis data mendapatkan seluruh token yang dimiliki dan nilai TF dari token tersebut terhadap lemma tersebut.
- d) Ulangi untuk semua item dalam variabel lemma:
- e) Ulangi untuk semua token yang terdapat pada item:
- f) $TFIDF_{item,token} = TF_{item,token} * IDF_{item}$
- g) Jika semua token pada perulangan (e) sudah diiterasi lanjutkan ke (h) jika tidak kembali ke (e)
- h) Jika semua item pada perulangan (d) sudah diiterasi lanjutkan ke (i) jika tidak kembali ke (d)
- i) Simpan hasil TFIDF ke dalam basis data.

Sehingga kita mendapat nilai TFIDF sebagai berikut.

$$TF_{warta} = \begin{bmatrix} berita, 0.17609 \\ kabar, 0.17609 \end{bmatrix}$$

$$TF_{informasi} = \begin{bmatrix} tahu, 0.47712 \\ kabar, 0.17609 \\ berita, 0.17609 \end{bmatrix}$$

G. Penghitungan Cosine Similarity antar Kata

Cosine similarity adalah metode untuk menghitung tingkat kemiripan dari dokumen. Cosine similarity memiliki rumu sebagai berikut :

$$\text{Cosine Similarity}(d1, d2) = \frac{\text{dotProduct}(d1, d2)}{\|d1\| * \|d2\|}$$

dimana d adalah dokumen. Kita perlu menerapkan perhitungan cosine similarity ke semua dokumen secara iteratif. Algoritma untuk menghitung cosine similarity sebagai berikut :

- Input : semua dokumen D yang telah dilakukan tokenisasi, remove punctuation, remove stop word, stemming.
- Output : nilai kemiripan antar dokumen
- Pseudocode :
- Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
 - Inisiasi array lemma dengan isi query mendapatkan semua lemma
 - Inisiasi array tfidf dengan index term kata dan isi dengan query dari basis data mendapatkan seluruh token yang dimiliki dan nilai TFIDF dari token tersebut terhadap lemma tersebut.
 - Ulangi untuk semua item1 dalam variabel lemma:
 - Ulangi untuk semua item2 dalam variabel lemma:
 - $\text{Cosine}(\text{item1}, \text{item2}) = \frac{\text{dotproduct}(\text{item1}, \text{item2})}{\|\text{item1}\| * \|\text{item2}\|}$
 - Jika semua token pada perulangan (e) sudah diiterasi lanjutkan ke (h) jika tidak kembali ke (e)

- h) Jika semua item pada perulangan (d) sudah diiterasi lanjutkan ke (i) jika tidak kembali ke (d)
- a. Simpan hasil Cosine ke dalam basis data.

dari algoritma diatas kita sudah mendapatkan nilai kemiripan antar dokumen yang mencerminkan semakin tinggi nilai kemiripan maka dapat dibilang kemungkinan antar dokumen menjadi sinonim semakin besar. Untuk contoh kasus diatas didapat nilai cosine similarity sebagai berikut :

$$\text{Cosine}(\text{warta}, \text{informasi}) = \frac{\text{dotProduct}(\text{warta}, \text{informasi})}{\|\text{warta}\| * \|\text{informasi}\|}$$

$$\begin{aligned} & \text{Cosine}(\text{warta}, \text{informasi}) \\ &= \frac{0.1769 * 0.1769 + 0.1769 * 0.1769 + 0 * 0.47712}{\sqrt{0.1769^2 + 0.1769^2} * \sqrt{0.1769^2 + 0.1769^2 + 0.47712^2}} \end{aligned}$$

$$\text{Cosine}(\text{warta}, \text{informasi}) = \frac{0.06201}{0.2501 * 0.5387} = \frac{0.06201}{0.1347}$$

$$\text{Cosine}(\text{warta}, \text{informasi}) = 0.4603$$

H. Filter Kandidat Sinonim Kata

Untuk menentukan kandidat sinonim dari suatu kata kita dapat menerapkan Thresholding pada hasil perhitungan cosine similarity diatas. Namun dengan adanya Thresholding ini akan ada kemungkinan muncul sebuah kata yang tidak memiliki sinonim dikarenakan semua nilai kemiripan dengan dokumen lain kurang dari nilai konstanta Threshold. Berikut algoritma untuk menerapkan Thresholding:

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array sinonim dengan isi query mendapatkan semua hasil cosine similarity
- c) Ulangi untuk semua item dalam variabel sinonim:

- d) Jika nilai cosine dari item $<$ threshold hapus item dari array sinonim.
- e) Jika semua item pada perulangan (c) sudah diiterasi lanjutkan ke (f) jika tidak kembali ke (c)
- f) Simpan hasil filtering ke dalam basis data.

3.2.4 Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus

Dalam pembentukan nilai kerjasama antar peneliti di ITS sebelumnya tidak memperhatikan sinonim kata yang berkaitan. Penghitungan topik antar peneliti dicocokkan secara umum tanpa melihat apakah antar kata yang dicocokkan merupakan sinonim satu sama lain. Misalkan dua kata yang diproses adalah kata “mutlak” dan “absolut” sebelumnya dua kata tersebut dianggap berbeda dalam perhitungan TF IDF. Padahal dalam praktisnya mutlak dan absolut merupakan sinonim kata, sehingga mutlak dan absolut perlu dianggap sebagai kata yang sama dalam perhitungan TF IDF agar mendapat perhitungan yang optimal.

A. Desain Umum

Untuk mendapatkan graph kerjasama peneliti dengan memperhatikan sinonim kata dalam pembentukannya maka perlu mengubah penghitungan pembentukan graph penelitiannya sedangkan algoritma k-means berbasis graphnya tidak ada yang perlu di modifikasi karena pengolahan kata hanya ada dalam pembentukan graph kerjasama peneliti ITS. Sehingga pada subbab ini hanya akan dijelaskan bagian yang perlu dirubah untuk menyesuaikan nilai kerjasama dengan sinonim kata tersebut. Bagian proses perlu di modifikasi tersebut berupa perhitungan TF topik, perhitungan IDF topik, perhitungan pembentukan graph kerjasama (cosine similarity) karena bagian tersebut merupakan bagian yang mengolah kata secara langsung.

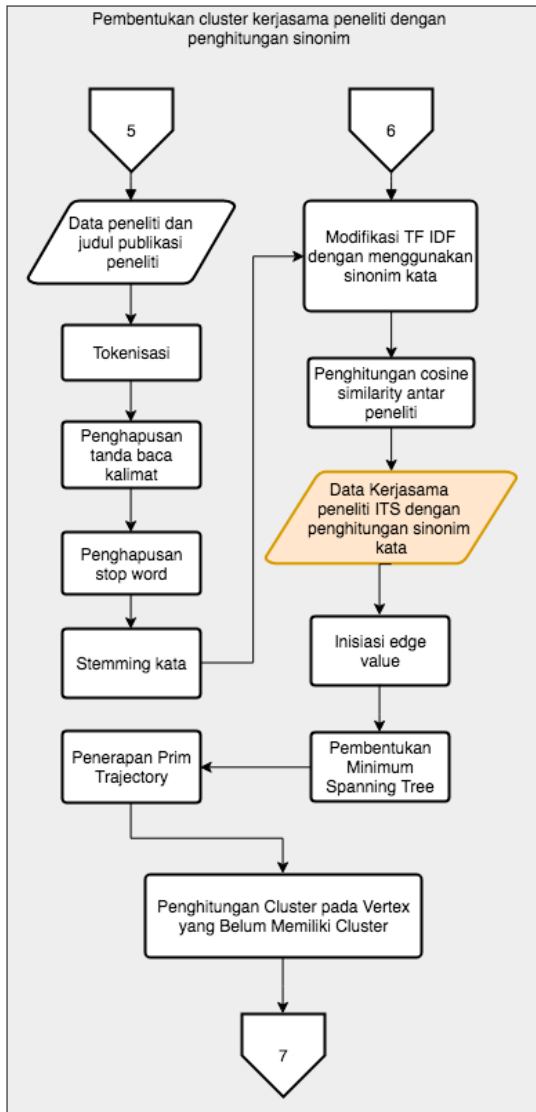
Dikarenakan secara keseluruhan proses dan alur pembentukan cluster kerjasama peneliti dengan perhitungan sinonim mirip dengan metode sebelumnya dan hanya beberapa proses saja yang diubah untuk menyesuaikan proses. Maka dalam subbab ini hanya proses yang dimodifikasi (dalam Gambar 3.9. **Pembentukan cluster kerjasama peneliti dengan perhitungan sinonim** Proses yang berwarna biru) saja yang dijelaskan. Selebihnya mengacu pada bab sebelumnya.

B. Data Peneliti dan Judul Publikasi Peneliti

Pada pembentukan graph kerjasama peneliti ITS data yang digunakan adalah data peneliti dan judul publikasi peneliti. Data peneliti nantinya akan menjadi identifier dengan dokumen nya adalah semua judul publikasi yang dimiliki oleh peneliti tersebut. Data peneliti dan publikasi peneliti merupakan data salinan dari basis data Pusat Data Terintegrasi LPTSI ITS. Sehingga sebelum melakukan praproses data atau pembobotan perlu dilakukan penyesuaian dengan data sehingga semua judul penelitian seorang peneliti berada dalam satu baris data. Misalkan kita memiliki data seperti pada Tabel 3.4:

Tabel 3.4 Tabel publikasi peneliti dosen ITS

id	Kode_peneliti	Judul_publikasi
1	P1	JP1
2	P2	JP2
3	P1	JP3
4	P3	JP4
5	P1	JP5
6	P2	JP6
..



Gambar 3.9. Pembentukan cluster kerjasama peneliti dengan perhitungan sinonim

Dalam pembentukan graph kerjasama peneliti ITS perlu dilakukan penyesuaian data publikasi seperti Tabel 3.2-4 dengan melakukan mengelompokkan judul penelitian sesuai dengan kode peneliti sehingga didapat table untuk dihitung bobotnya seperti pada Tabel 3.5:

Tabel 3.5 Tabel distinct publikasi peneliti dosen ITS

id	Kode_peneliti	Judul_publikasi
1	P1	JP1, JP3, JP5
2	P2	JP2, JP6
3	P3	JP3
..

Dengan bentuk data seperti diatas maka judul publikasi per peneliti siap untuk dilakukan pengolah teks dan penghitungan bobot kerjasama.

C. Modifikasi penghitungan TF

Pada umumnya nilai TF dihitung tanpa memperhatikan sinonim dari dua kata, misalkan dua kata yang diproses adalah kata “mutlak” dan “absolut” sebelumnya dua kata tersebut dianggap berbeda dalam perhitungan TF. Padahal dalam praktisnya mutlak dan absolut merupakan sinonim kata, sehingga mutlak dan absolut perlu dianggap sebagai kata yang sama dalam perhitungan TF agar mendapat perhitungan yang optimal.

Sehingga apabila kita mempunyai sebuah mempunyai dokumen publikasi peneliti yang telah dilakukan tahap praproses seperti dibawah ini:

$$Token_{peneliti1} = \begin{bmatrix} "sistem" \\ "kompresor" \\ "mesin" \\ "motor" \\ "regenerative" \\ "sistem" \end{bmatrix}$$

$$Token_{peneliti2} = \begin{bmatrix} "sistem" \\ "mesin" \\ "dinamis" \\ "gelombang" \end{bmatrix}$$

Dan sebuah sinonim kata sebagai berikut :

$$Sinonim = \begin{bmatrix} mesin = motor \\ mesin = kompresor \end{bmatrix}$$

Dalam penghitungan TF suatu kata dalam dokumen perlu dicek apakah didalam dokumen tersebut juga ada sinonim dari kata tersebut. Jika kata tersebut dianggap dalam satu dokumen maka TF kata tersebut ditambahkan dengan TF dari TF kata sinonimnya. Seperti kata mesin, motor dan kompresor adalah sinonim satu sama lain. Berikut langkah-langkah yang dilakukan untuk modifikasi penghitungan nilai TF:

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array lemma dengan index lemma dan isi dengan semua token yang telah di stemming.
- c) Inisiasi array sinonim yang berisi kata dan sinonimnya
- d) Ulangi untuk semua item dalam variabel lemma:
- e) Ulangi untuk semua token yang terdapat pada item:
- f) $F_{item,token}++$
- g) Jika $F_{item,token}$ terdapat pada variable sinonim maka $F_{item,token} = F_{item,token} + sinonim_{token}$

- h) Jika semua token pada perulangan (e) sudah diiterasi lanjutkan ke (i) jika tidak kembali ke (e)
- i) Ulangi untuk semua nilai yang terdapat pada variabel F:
- j) $TF_{item,token} = 1 + \log_{10} F_{item,token}$
- k) Jika semua nilai F pada perulangan (i) sudah diiterasi lanjutkan ke (l) jika tidak kembali ke (i)
- l) Jika semua item pada perulangan (d) sudah diiterasi lanjutkan ke (m) jika tidak kembali ke (d)
- m) Simpan hasil TF ke dalam basis data.

Sehingga untuk modifikasi penghitungan TF menghasilkan nilai TF sebagai berikut:

$$F_{peneliti1} = \begin{bmatrix} sistem = 2 \\ \text{kompresor} = 1 \\ mesin = 1 \\ motor = 1 \\ regenerative = 1 \end{bmatrix} \xrightarrow{\text{sinonim}} \begin{bmatrix} sistem = 2 \\ kompresor = 3 \\ mesin = 3 \\ motor = 3 \\ regenerative = 1 \end{bmatrix}$$

$$TF_{peneliti1} = \begin{bmatrix} sistem = 1,301 \\ kompresor = 1,4771 \\ mesin = 1,4771 \\ motor = 1,4771 \\ regenerative = 1 \end{bmatrix},$$

$$F_{peneliti2} = \begin{bmatrix} sistem = 1 \\ mesin = 1 \\ dinamis = 1 \\ gelombang = 1 \end{bmatrix} \xrightarrow{\text{sinonim}} \begin{bmatrix} sistem = 1 \\ mesin = 1 \\ dinamis = 1 \\ gelombang = 1 \end{bmatrix}$$

$$TF_{peneliti2} = \begin{bmatrix} sistem = 1 \\ mesin = 1 \\ dinamis = 1 \\ gelombang = 1 \end{bmatrix}$$

D. Modifikasi penghitungan IDF

Begitu juga dengan nilai IDF perlu dilakukan penghitungan ulang yang dimodifikasi dengan memperhatikan sinonim antar kata seperti disampaikan diatas. Sehingga apabila kita mempunyai sebuah mempunyai dokumen publikasi peneliti yang telah dilakukan tahap praproses seperti dibawah ini:

$$Token_{peneliti1} = \begin{bmatrix} "sistem" \\ "kompresor" \\ "mesin" \\ "motor" \\ "regenerative" \\ "sistem" \end{bmatrix}$$

$$Token_{peneliti2} = \begin{bmatrix} "sistem" \\ "mesin" \\ "dinamis" \\ "gelombang" \end{bmatrix}$$

Sama dengan penghitungan TF diatas modifikasi penghitungan IDF juga harus menambahkan nilai IDF dari kata yang merupakan sinonim dari kata tersebut hingga nilai DF tidak melebihi total dokumen peneliti yang ada. Berikut langkah-langkah yang harus di lakukan untuk menghitung IDF yang telah dimodifikasi dengan penghitungan sinonim kata:

- a) Impor pustaka python untuk menghubungkan python dengan basis data SQLServer, pymssql.
- b) Inisiasi array term dengan index term dan isi dengan nilai query ke basis data banyaknya term muncul dalam semua dokumen.
- c) Inisiasi array sinonim yang berisi kata dan sinonimnya
- d) Ulangi untuk semua item dalam variabel lemma:
- e) Jika item terdapat pada variable sinonim maka lemma[item] += lemma[sinonim[item]]
- f) $IDF_{item} = \log_{10} (\text{Jumlah dokumen} + 1) / lemma[item]$

- g) Jika semua item pada perulangan (d) sudah diiterasi lanjutkan ke (h) jika tidak kembali ke (d)
- h) Simpan hasil IDF ke dalam basis data.

Sehingga penghitungan modifikasi IDF akan menghasilkan nilai IDF sebagai berikut :

$$\begin{aligned}
 \mathbf{DF} &= \begin{bmatrix} \text{sistem} = 2 \\ \text{kompresor} = 1 \\ \text{mesin} = 2 \\ \text{motor} = 1 \\ \text{regenerative} = 1 \\ \text{dinamis} = 1 \\ \text{gelombang} = 1 \end{bmatrix} \xrightarrow{\text{sinonim}} \begin{bmatrix} \text{sistem} = 2 \\ \text{kompresor} = 2 \\ \text{mesin} = 2 \\ \text{motor} = 2 \\ \text{regenerative} = 1 \\ \text{dinamis} = 1 \\ \text{gelombang} = 1 \end{bmatrix} \\
 \mathbf{IDF} &= \begin{bmatrix} \text{sistem} = 0.176 \\ \text{kompresor} = 0.176 \\ \text{mesin} = 0.176 \\ \text{motor} = 0.176 \\ \text{regenerative} = 0.477 \\ \text{dinamis} = 0.477 \\ \text{gelombang} = 0.477 \end{bmatrix}
 \end{aligned}$$

Tahapan selanjutnya adalah pembobotan kerjasama peneliti di ITS. Tahapan ini sama dengan metode sebelumnya pada Subbab 3.2.3. dan pembentukan graph kerjasama antar peneliti sama dengan subbab 3.2.2.

BAB IV IMPLEMENTASI

Bab ini membahas proses implementasi yang dilakukan berdasarkan hasil perancangan pada Bab III. Penjelasan proses implementasi terbagi menjadi enam bagian, yaitu lingkungan implementasi, implementasi basis data, implementasi K-Means berbasis graph, implementasi pembentukan sinonim thesaurus kateglo, expand pemnbentukan graph kerjasama dengan sinonim thesaurus, dan penggunaan kerangka kerja dalam visualisasi hasil clustering.

4.1 Lingkungan Implementasi

Lingkungan Implementasi adalah lingkungan di mana sistem tugas akhir ini dibangun. Lingkungan implementasi dibagi dua yaitu perangkat keras dan perangkat lunak.

4.1.1 Perangkat Keras

Lingkungan implementasi perangkat keras dari tugas akhir ini adalah sebagai berikut:

- Tipe : PC Gygabyte Technology
- Prosesor : Inter® Core(TM) i5-3330 CPU (4 CPUs)
@ 3.2GHz
- Memori (RAM): 8192MB RAM

4.1.2 Perangkat Lunak

Lingkungan implementasi perangkat lunak dari tugas akhir ini adalah sebagai berikut:

- Sistem operasi : Windows 10 Enterprise 64-bit
- Bahasa Pemrograman : PHP 5.6.3, Python 2.7.11
- *Text Editor* : PHPStorm 2016.1.2, PyCharm 2016.1

- Kerangka Kerja : Laravel 5.2
- *Web Server* : Apache 2.4.4
- Basis Data : MySQL 5.5.32, SQL Server 2008
- Kakas Bantu Basis Data : Navicat Premium 11.1.7 (64-bit)

4.2 Implementasi Basis data

Dalam penyusunan tugas akhir ini data diambil dari beberapa sumber basis data. Seperti daftar kata stop word pada tugas akhir ini menggunakan 2 sumber stop word, data kerjasama peneliti di ITS, dan basis data kateklo untuk pembentukan sinonim kata.

4.2.1 Daftar Stop Word

Pada pengerjaan tugas akhir ini terdapat dua proses yang menggunakan daftar stop word yaitu penghapusan stop word dari dokumen definisi kata pada basis data kateklo dan penghapusan stop word dari dokumen judul penelitian. Pada dokumen definisi kata kateklo bisa dikatakan seluruh kata yang menyusun adalah kata umum yang sering digunakan sehingga tidak perlu stop word khusus dalam praproses datanya, sedangkan untuk kata-kata dalam dokumen judul penelitian kebanyakan mengandung kata ilmiah yang sama, padahal kata ilmiah tersebut tidak termasuk dalam daftar stop word bahasa Indonesia. Keterangan stopwords tersebut dapat dilihat dalam Tabel 4.1.

Tabel 4.1 Tabel keterangan stopwords

Keterangan	Stop Word Dokumen Definisi Kata Kateklo	Stop Word Dokumen Judul Publikasi Peneliti
Sumber	Computer Science Dept. Universitas Brawijaya [5]	Stop word Sistem Informasi Repositori Peneliti ITS
Jumlah	357 kata	2441 kata
Bahasa	Bahasa Indonesia	Bahasa Indonesia dan Bahasa Inggris


Stop word dokumen judul publikasi peneliti membutuhkan stop word dua bahasa karena judul penelitian juga ada yang berbahasa inggris. Pada stop word dokumen definisi kata kataglo merupakan standard stop word bahasa indonesia yang biasa digunakan dalam pengolahan teks seperti lalu, selama, nasih, maupun, manakala, entah, dst. Stop word dokumen judul publikasi peneliti memiliki kandungan kata yang lebih kompleks. Kata-kata pada stop word tersebut memiliki kata yang umum dalam penelitian di ITS seperti kata institut, daya, solar, aspek, mahasiswa, ilmiah, jurusan, arsitektur, conference, international, dll. Untuk data lengkap stop word yang digunakan dalam tugas akhir ini terdapat pada tabel stop word di Lampiran.

4.2.2 Basis Data Repositori Peneliti ITS


Sistem informasi repositori peneliti ITS merupakan sebuah sistem informasi untuk menampilkan publikasi dan penelitian yang dimiliki oleh ITS. Basis data sistem informasi repositori peneliti menggunakan SQL Server 2008 dan implementasi koneksi ke database SQL Server menggunakan Navicat dan pustaka pymssql dalam bahasa pemrograman python. Graph kerjasama peneliti ITS juga terdapat pada sistem informasi repositori peneliti ITS. Berikut tabel pada basis data repositori peneliti di ITS yang digunakan dalam tugas akhir ini. Gambar 4.1 – Gambar 4.3 dan Tabel 4.2 merupakan informasi database yang digunakan.

Name	Type	Length	Scale	Not null	
bobot_peneliti	float	53	0	<input type="checkbox"/>	
bobot_topik	float	53	0	<input type="checkbox"/>	
final	float	53	0	<input type="checkbox"/>	
kode_pegawai1	int	0	0	<input type="checkbox"/>	
kode_pegawai2	int	0	0	<input type="checkbox"/>	

Gambar 4.1 Tabel tran_bobot_dosen_peneliti

Name	Type	Length	Scale	Not null	
kode	int	0	0	<input checked="" type="checkbox"/>	 1
kode_publikasi	varchar	255	0	<input type="checkbox"/>	
kode_kegiatan_publikasi	varchar	2	0	<input type="checkbox"/>	
kode_jenis_peneliti	varchar	1	0	<input type="checkbox"/>	
kode_media_publikasi	varchar	1	0	<input type="checkbox"/>	
kode_pelaksanaan_penelitian	varchar	1	0	<input type="checkbox"/>	
kode_jenis_pembiayaan	varchar	1	0	<input type="checkbox"/>	
kode_status_validasi	varchar	2	0	<input type="checkbox"/>	
kode_periode_pelaporan	varchar	4	0	<input type="checkbox"/>	
kode_pegawai	varchar	255	0	<input type="checkbox"/>	
jumlah_pembiayaan	float	53	0	<input type="checkbox"/>	
tahun	varchar	4	0	<input type="checkbox"/>	
bulan	varchar	2	0	<input type="checkbox"/>	
judul	varchar(MAX)	-1	0	<input type="checkbox"/>	
kata_kunci	varchar	255	0	<input type="checkbox"/>	
total_waktu	int	0	0	<input type="checkbox"/>	
lokasi	varchar	255	0	<input type="checkbox"/>	
kode_log_audit	varchar	255	0	<input type="checkbox"/>	
abstraksi	varchar(MAX)	-1	0	<input type="checkbox"/>	
pengarang	varchar	255	0	<input type="checkbox"/>	
keterangan	varchar	255	0	<input type="checkbox"/>	
deleted_at	datetime	0	0	<input type="checkbox"/>	

Gambar 4.2 Tabel tran_publikasi_dosen_tetap

Name	Type	Length	Scale	Not null	
kode_fakultas	char	2	0	<input type="checkbox"/>	
kode_jurusan	char	5	0	<input type="checkbox"/>	
nama_dosen	char	100	0	<input type="checkbox"/>	
gelar_akademik_depan	varchar	30	0	<input type="checkbox"/>	
gelar_akademik_belakang	varchar	30	0	<input type="checkbox"/>	
skor_spmi	float	53	0	<input type="checkbox"/>	
kualifikasi_ahli	varchar(MAX)	-1	0	<input type="checkbox"/>	
nama_jurusan_en	varchar	255	0	<input type="checkbox"/>	
journals	int	0	0	<input type="checkbox"/>	
conferences	int	0	0	<input type="checkbox"/>	
books	int	0	0	<input type="checkbox"/>	
kode_pegawai	int	0	0	<input type="checkbox"/>	
thesis	int	0	0	<input type="checkbox"/>	
paten	int	0	0	<input type="checkbox"/>	
research	int	0	0	<input type="checkbox"/>	
id_node	int	0	0	<input type="checkbox"/>	
nip	varchar	18	0	<input type="checkbox"/>	
nama_fakultas_en	varchar	255	0	<input type="checkbox"/>	
email	varchar	255	0	<input type="checkbox"/>	
id	int	0	0	<input checked="" type="checkbox"/>	 1
best_term	varchar(MAX)	-1	0	<input type="checkbox"/>	

Gambar 4.3 Tabel tran_temp_dosen

Tabel 4.2 Keterangan tabel hasil duplikasi data dari basis data peneliti ITS

tran_publikasi_dosen_tetap	
Keterangan	Tabel tran_publikasi_dosen_tetap merupakan tabel yang menyimpan informasi tentang publikasi yang dimiliki peneliti(dosen) di ITS
Jumlah Data	25.261
Kolom yang digunakan	kode_pegawai, judul
tran_temp_dosen	
Keterangan	Tabel tran_temp_dosen merupakan tabel yang menyimpan informasi tentang peneliti (dosen) di ITS
Jumlah Data	983
Kolom yang digunakan	kode_pegawai, nama_dosen
tran_bobot_dosen_peneliti	
Keterangan	Tabel tran_bobot_dosen_peneliti merupakan tabel yang menyimpan informasi tentang bobot kerjasama antar peneliti(dosen) di ITS. Tabel inilah yang menyimpan graph kerjasama antar peneliti
Jumlah Data	460.361
Kolom yang digunakan	Semua kolom


4.2.3 Kateglo

Pada basis data kateglo terdapat beberapa tipe kata atau leksikal. Dalam pembentukan sinonim dengan basis data kateglo

tidak semua data lemma diproses, lemma yang diproses hanya lemma yang mempunyai leksikal nomina(kata benda) dan verba(kata kerja) dan yang memiliki tipe kata dasar. Filter kata dasar diperlukan karena pembobotan kata pada Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus kata yang dibobot sudah diterapkan dalam kata dasar melalui metode stemming. Sehingga implementasi basis data pada basis data katego sebagai berikut. Gambar 4.4, Gambar 4.5 dan Tabel 4.3 merupakan struktur dan informasi tabel yang dimiliki oleh katego.

Name	Type	Length	Scale	Not null	
id	int	0	0	<input type="checkbox"/>	
lemma	varchar	1024	0	<input type="checkbox"/>	
lexical_id	int	0	0	<input type="checkbox"/>	
definition	varchar(MAX)	-1	0	<input type="checkbox"/>	

Gambar 4.4 Tabel proc_definition

Name	Type	Length	Decimals	Not null	
lexical_id	int	11	0	<input checked="" type="checkbox"/>	 1
lexical_name	varchar	255	0	<input checked="" type="checkbox"/>	
lexical_abbreviation	varchar	255	0	<input checked="" type="checkbox"/>	

Gambar 4.5 Tabel lexical

Tabel 4.3 Keterangan tabel katego

Tabel	Keterangan	Jumlah Data
proc_defi nition	Tabel proc_defintion menyimpan lemma, definisi dan tipe lexicalnya	33.189
lexical	Tabel lexical menyimpan tipe sebuah kata	2

4.3 Implementasi K-Means Berbasis Graph

Tahapan implementasi dari hasil proses perancangan

yang telah dilakukan pada Bab III. Seluruh tahapan implementasi K-Means berbasis graph sesuai dengan perancangan pada Bab III menggunakan bahasa pemrograman python.

4.3.1 Inisiasi Edge Value

Implementasi inisiasi nilai edge dilakukan dengan menggunakan bahasa pemrograman python dan pustaka pymssql untuk melakukan query ke basis data. Berikut potongan kode pengubahan nilai bobot kerjasama ke dalam nilai jarak edge.

```
import pymssql
from decimal import *
...
source_cursor.execute('SELECT MAX(value) FROM
edge')
all_data = source_cursor.fetchall()
max = all_data[0][0]

source_cursor.execute('SELECT * FROM edge')
all_data = source_cursor.fetchall()
for data in all_data:
    value = float(max-data[1])
    source_cursor.execute("UPDATE edge SET value =
%d WHERE id = %s", ((value,data[0])))
...
```

Kode Sumber 4.1 Proses pengubahan nilai bobot kerjasama ke nilai edge

Pada Kode Sumber 4.1 terdapat kode pengubahan nilai bobot kerjasama ke dalam nilai jarak edge. Pertama kali yang dilakukan dalam kode tersebut membuka koneksi ke database. Selanjutnya mendapatkan nilai maksimal dari nilai edge. Setelah itu query ke database untuk mendapatkan semua edge untuk dihitung kembali nilai edgenya. Setelah mendapat nilai dalam variabel value dilakukan pemanggilan fungsi update untuk mengupdate ke dalam basis data nilai edge.

4.3.2 Minimum Spanning Tree

Implementasi pembentukan minimum spanning tree dari data kerjasama peneliti menggunakan bahasa pemrograman python dan pustaka pymysql untuk melakukan sambungan data ke basis data. Kode Sumber 4.2 merupakan potongan kode untuk pembentukan minimum spanning tree.

```
import pymysql

...
peneliti=[]
mst = []

source_cursor.execute('SELECT TOP 1 kode_peneliti1
FROM edge')
init_node = source_cursor.fetchone()

query = "SELECT * FROM edge WHERE kode_peneliti1 =
'{}' OR kode_peneliti2 =
'{}'".format(init_node[0],init_node[0])
source_cursor.execute(query)
edges = source_cursor.fetchall()
edges = sorted(edges,key=lambda edge: edge[1])
peneliti[init_node[0]] = 1

while len(edges) > 0:
    if peneliti[edges[0][2]] == 0 or
peneliti[edges[0][3]] == 0:
        mst.append((edges[0][0],edges[0][1],
edges[0][2],edges[0][3]))
        tmp_edges = []
        if peneliti[edges[0][2]] == 0:
            query = "SELECT * FROM edge WHERE
kode_peneliti1 = '{}' OR kode_peneliti2 =
'{}'".format(edges[0][2],edges[0][2])
            source_cursor.execute(query)
            tmp_edges = source_cursor.fetchall()
        if peneliti[edges[0][3]] == 0:
            query = "SELECT * FROM edge WHERE
kode_peneliti1 = '{}' OR kode_peneliti2 =
'{}'".format(edges[0][3],edges[0][3])
            source_cursor.execute(query)
            tmp_edges = source_cursor.fetchall()
```

```

        peneliti[edges[0][2]] = 1
        peneliti[edges[0][3]] = 1
        for tmp in tmp_edges:
            if peneliti[tmp[2]] == 0 or
peneliti[tmp[3]] == 0:
                edges.append(tmp)
            edges = sorted(edges, key=lambda edge:
edge[1])
            edges.pop(0)

source_cursor.execute("DELETE FROM step_mst")
source_cursor.executemany("INSERT INTO
step_mst(id,value,kode_peneliti1,kode_peneliti2)
VALUES (%d, %d, %s, %s)",mst)
source_conn.commit()
source_conn.close()

```

Kode Sumber 4.2 Proses Pembentukan Minimum Spanning Tree

Variabel `init_node` merupakan variabel yang digunakan untuk menyimpan vertex initial dari tabel `edge`. Lalu dilakukan pengambilan semua `edge` yang dimiliki oleh vertex `init_node` tersebut yang disimpan dalam `queue` dan hasilnya disorting secara menaik. Lalu langkah sebelumnya diulangi hingga semua `edges` diproses. Dari kode sumber diatas dan data kerjasama peneliti di ITS didapatkan 958 `edge` dan 959 vertex peneliti.

4.3.3 Penerapan Prim Trajectory

Implementasi penerapan `prim trajectory` dari data kerjasama peneliti yang telah diterapkan `minimum spanning tree` menggunakan bahasa pemrograman `python` dan pustaka `pymssql` untuk melakukan sambungan data ke basis data. Berikut potongan kode untuk penerapan `prim trajectory`. Kode Sumber 4.3 merupakan potongan kode penerapan `prim trajectory`. Tabel 4.4 merupakan tabel hasil penerapan `prim trajectory` dan mendapatkan beberapa nilai `K`.

```

import pymysql
...
peneliti = []
queue = []
flag = []
trajectory = []

query = "SELECT * FROM step_mst WHERE value < ".C
source_cursor.execute(query)
edges = source_cursor.fetchall()
edges = sorted(edges, key=lambda edge: edge[1])
cluster = 1

for edge in edges:
    if flag[peneliti.index(edge[2])] == 0:
        queue.append(edge[2])
        count = 0
        tmp_trajectory = []
        tmp_trajectory.append((edge[0], edge[1],
edge[2], edge[3], cluster))
        while len(queue) > 0:
            count+=1
            tmp_peneliti = queue[0]
            flag[peneliti.index(tmp_peneliti)] = 1;
            queue.pop(0)
            query = "SELECT * FROM step_mst WHERE
value < ".C." AND (kode_peneliti1 = '{} ' OR
kode_peneliti2 =
 '{} ')".format(tmp_peneliti, tmp_peneliti)
            source_cursor.execute(query)
            tmp_edges = source_cursor.fetchall()
            for iterator in tmp_edges :
                if tmp_peneliti == iterator[2] and
flag[peneliti.index(iterator[3])] == 0:
                    queue.append(iterator[3])
                    if tmp_peneliti == iterator[3] and
flag[peneliti.index(iterator[2])] == 0:
                        queue.append(iterator[2])
                        if (iterator[0], iterator[1],
iterator[2], iterator[3], cluster) not in
tmp_trajectory:

```

```

        tmp_trajectory.append((iterator[0],
iterator[1], iterator[2], iterator[3], cluster))
        if count > K:
            trajectory += tmp_trajectory
            cluster += 1

source_cursor.execute("DELETE FROM step_prim")
source_cursor.executemany("INSERT INTO
step_prim(id,value,kode_peneliti1,kode_peneliti2,cl
ass) VALUES (%d, %d, %s, %s, %d)",trajectory)

source_conn.commit()
source_conn.close()

```

Kode Sumber 4.3 Proses Penerapan Prim Trajectory

Variabel peneliti berisi semua kode_peneliti dari tabel step_mst. Dan variabel trajectory adalah variabel untuk menyimpan hasil dari kode tersebut. Yang perlu diperhatikan adalah variabel C dan N yang berdampak sekali pada hasil cluster awal apabila dilakukan perubahan. Nilai C dalam tugas akhir ini ditentukan dengan melakukan pengecekan ke dalam baris yang terambil pada tabel edge sehingga jumlah baris yang dikembalikan kurang dari setengah jumlah bari dari basis data, sehingga didapatkan nilai 0.84 sebagai nilai dari C dengan jumlah baris edge yang didapatkan adalah 384 data edge. Nilai N merupakan jumlah vertex minimal yang terbentuk untuk suatu sub cluster dimasukkan kedalam suatu class. Berikut tabel hasil pengubahan nilai C dan K.

Tabel 4.4 Tabel Pengubahan Nilai Konstanta C dan K

Nilai C	Nilai K	Cluster awal yang terbentuk
0.84	15	7
0.84	10	10
0.84	5	16

4.3.4 Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster

Implementasi perhitungan cluster pada vertex yang belum memiliki cluster menggunakan bahasa pemrograman python dan pustaka pymssql untuk melakukan sambungan data ke basis data. Kode Sumber 4.4 merupakan potongan kode untuk perhitungan cluster pada vertex yang belum memiliki cluster.

```
import pymssql
...
incluster = []
outcluster = []
peneliti = []
queue = []
flag = []

for outer in incluster:
    for inner in outer:
        flag[peneliti.index(inner)] = 1

for i, outer in enumerate(incluster):
    for inner in outer:
        query = "SELECT * FROM step_mst WHERE
kode_peneliti1 = '{} ' OR kode_peneliti2 =
'{}' ".format(inner, inner)
        source_cursor.execute(query)
        tmp_edges = source_cursor.fetchall()
        for iterator in tmp_edges:
            if inner == iterator[2] and
flag[peneliti.index(iterator[3])] == 0:
                queue.append(iterator)
            if inner == iterator[3] and
flag[peneliti.index(iterator[2])] == 0:
                queue.append(iterator)
queue = sorted(queue, key=lambda edge: edge[1])
hasil = []
while len(queue) > 0:
    if flag[peneliti.index(queue[0][2])] == 0 or
flag[peneliti.index(queue[0][3])] == 0:
```



```

        if flag[peneliti.index(queue[0][3])] == 0:
            query = "SELECT * FROM step_mst WHERE
kode_peneliti1 = '{}' OR kode_peneliti2 =
'{}'".format(queue[0][3], queue[0][3])
            source_cursor.execute(query)
            tmp_edges = source_cursor.fetchall()
            for iterator in tmp_edges:
                queue.append(iterator)
        if flag[peneliti.index(queue[0][2])] == 0:
            query = "SELECT * FROM step_mst WHERE
kode_peneliti1 = '{}' OR kode_peneliti2 =
'{}'".format(queue[0][2], queue[0][2])
            source_cursor.execute(query)
            tmp_edges = source_cursor.fetchall()
            for iterator in tmp_edges:
                queue.append(iterator)
        flag[peneliti.index(queue[0][2])] = 1
        flag[peneliti.index(queue[0][3])] = 1
        hasil.append(queue[0])
    queue.pop(0)
    queue = sorted(queue, key=lambda edge: edge[1])

hasil = sorted(hasil, key=lambda edge: edge[4])
source_cursor.execute("DELETE FROM step_final_K16")
source_cursor.executemany("INSERT INTO
step_final_K16(id,value,kode_peneliti1,kode_penelit
i2,class) VALUES (%d, %d, %s, %s, %d)", hasil)

source_conn.commit()
source_conn.close()

```

Kode Sumber 4.4 Proses Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster

Setelah kode diatas dijalankan maka seluruh vertex yang belum memiliki cluster akan dikelompokkan sesuai dengan cluster terdekat. Variabel `incluster` menyimpan semua vertex yang telah berada dalam suatu kelas sedangkan variabel `outcluster` menyimpan vertex yang belum memiliki kelas. Variabel `peneliti` menyimpan semua `kode_peneliti` dalam tabel `step_mst`.

4.4 Implementasi Pembentukan Sinonim Thesaurus Kateglo

Tahapan implementasi dari hasil proses perancangan Pembentukan Sinonim Thesaurus Kateglo yang telah dilakukan pada Bab III. Seluruh tahapan implementasi Pembentukan Sinonim Thesaurus Kateglo sesuai dengan perancangan pada Bab III menggunakan bahasa pemrograman python.

4.4.1 Praproses Data

Implementasi praproses data menggunakan bahasa pemrograman python dan pustaka sesuai dengan perancangan pada Bab III. Praproses data yang dilakukan pada data definisi lemma adalah tokenisasi, penghapusan tanda baca kalimat, penghapusan stop word, dan stemming kata. Koneksi ke basis data menggunakan pustaka mysql.connector karena basis data kateglo menggunakan MySQL sebagai DBMS. Berikut implementasi setiap langkah pada praproses data.

A. Tokenisasi

```
import mysql.connector
from nltk.tokenize import word_tokenize
...
    token = word_tokenize(definition)
...
```

Kode Sumber 4.5 Implementasi Tokenisasi

Tokenisasi diimplementasikan pada awal setiap kali pembacaan data dokumen definisi lemma. Implementasi tokenisasi menggunakan pustaka python dari nltk yaitu word_tokenize. Setelah dilakukan tokenisasi pada kata variabel definition maka hasil dari tokenisasi disimpan dalam variabel token untuk diproses pada proses selanjutnya. Kode Sumber 4.5 merupakan potongan kode untuk implementasi tokenisasi pada dokumen.

B. Penghapusan Tanda Baca Kalimat

```
import mysql.connector
import string
...
    unicode_token = [s.encode('ascii') for s
in token]
    punct = filter(lambda x: x not in
string.punctuation, unicode_token)
...
```

Kode Sumber 4.6 Implementasi Penghapusan Tanda Baca Kalimat

Penghapusan tanda baca kalimat pada python perlu dilakukan dalam dua proses yaitu mengubah format teks dari unicode ke dalam tipe string lalu melakukan pengecekan ke setiap karakter dalam string tersebut. Implementasi penghapusan tanda baca ini menggunakan pustaka string dalam bahasa pemrograman python. Kode Sumber 4.6 merupakan potongan kode untuk penghapusan tanda baca kalimat.

C. Penghapusan Stop Word

```
import mysql.connector
...
    stop_words = set()
    with open("stop.txt", "r") as ins:
        for line in ins:
            stop_words.add(line.rstrip('\n'))
    stopped = filter(lambda x: x not in
stop_words, punct)
...
```

Kode Sumber 4.7 Implementasi Penghapusan Stop Word

Pada Kode Sumber 4.7 terdapat kode untuk melakukan proses membaca isi berkas stop.txt yang berisi stop word dan memasukkannya ke dalam variabel stop_words. Lalu variabel string definisi yang diproses adalah variabel hasil proses

penghapusan tanda baca. Setelah selesai dilakukan stop word pada keseluruhan kata maka hasil penghapusan stop word di simpan dalam variabel `stopped`.

D. Stemming Kata

```
import mysql.connector
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

...
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    cleaned_text = map(lambda x:
        stemmer.stem(x), stopped)
...
```

Kode Sumber 4.8 Implementasi Stemming Kata

Implementasi stemming kata menggunakan pustaka python Sastrawi. Sastrawi merupakan pustaka pengolahan bahasa indonesia yang ada dalam bahasa python. Kata yang diolah dalam tahap stemming adalah kata yang sudah dihilangkan stop wordnya. Hasil dari implementasi kata langsung diolah untuk mendapatkan nilai TF dari masing-masing kata terhadap definisi kata. Kode Sumber 4.8 merupakan potongan kode untuk stemming kata.

4.4.2 Pembobotan Kata

A. Penghitungan Term Frequency

```
for (id, lemma, lexical_id, definition) in
data:
...
    for s in cleaned_text:
        if s not in index:
            index.append(s)
            value[index.index(s)] += 1
        for s in index:
            res = float(float( value[
index.index(s)] ) / len(cleaned_text))
            if res >= 0.1 :
```

```
data_insert.append((s, id,
value[index.index(s)],len(cleaned_text),res))
```

Kode Sumber 4.9 Penghitungan Term Frequency

Implementasi penghitungan TF dari setiap kata menggunakan bahasa python dan hasilnya disimpan dalam basis data dalam bentuk baris record untuk masing-masing kata definisi dalam lemma. Kode Sumber 4.9 merupakan potongan kode untuk menghitung nilai TF.

B. Penghitungan Inverse Document Frequency

```
import mysql.connector
import math
...
data = []
data_insert = []
iter = 0
for (term,count_item) in data:
    iter += 1
    idf = math.log10(float(count)/ count_item)
    data_insert.append((term, count_item,idf))
cursor.close()
cnx.close()
```

Kode Sumber 4.10 Penghitungan Inverse Document Frequency

Implementasi IDF pada tugas akhir ini menggunakan bahasa python dan pustaka math untuk mendapatkan nilai log suatu angka. Variabel data berisi semua data dalam bentuk struct (lemma,count_item). Variabel count_item merupakan hasil query dalam basis data yang mengembalikan jumlah data yang muncul dalam semua dokumen. Variabel count merupakan hasil query dalam basis data yang mengembalikan seluruh jumlah dokumen yang diproses. Setelah mendapatkan nilai idf seluruh data dimasukkan ke dalam basis data. Kode Sumber 4.10 merupakan potongan kode untuk penghitungan IDF.

C. Penghitungan Bobot Kata TF IDF

```
import mysql.connector
import math

def get_idf():
    ...
    for (id,term, doc_term , result) in cursor:
        idf[str(term)] = result
    cursor.close()
    cnx.close()
    return idf
```

Kode Sumber 4.11 Fungsi Mendapatkan Nilai IDF

```
def calculate():
    ...
    idf = get_idf()
    data = []
    data_insert = []
    iter = offset+1
    for (term, id_proc_definition, result) in data:
        iter += 1
        tfidf = result*idf[str(term)]
    ...
```

Kode Sumber 4.12 Fungsi Menghitung nilai TFIDF

Implementasi penghitungan TFIDF menggunakan bahasa python. Fungsi `get_idf()` merupakan fungsi untuk mendapatkan semua nilai idf dari proses penghitungan sebelumnya. Variabel `data` merupakan variabel yang berisi semua hasil perhitungan TF pada proses sebelumnya. Sehingga perulangan `data` pada fungsi `calculate()` merupakan perulangan untuk semua nilai TF dan pada setiap perulangan dilakukan perulangan pada setiap nilai $TF * IDF$ dari term tersebut. Kode Sumber 4.11 dan Kode Sumber 4.12 merupakan kode implementasi dari penghitungan untuk mendapatkan nilai TF IDF dari suatu kata dalam dokumen.

D. Penghitungan Cosine Similarity Antar Lemma

```
import mysql.connector

def get_tfidf():
    tfidf = {}
    for (id, term, id_proc_definition , result) in
cursor:
        if str(id_proc_definition) not in tfidf:
            tfidf[str(id_proc_definition)] = {}
            tfidf[str(id_proc_definition)][str(term)] =
result
        return tfidf
```

Kode Sumber 4.13 Fungsi Mendapatkan Nilai TF IDF

```
def cosine(tfidf):
    data = []
    id = []
    lemma = {}
    for item in data:
        id.append(item[0])
        lemma[item[0]] = str(item[1])
    for idx1 in range(0, len(id)-1):
        for idx2 in range(idx1+1, len(id)):
            dotproduct = 0.0
            if id[idx1] != id[idx2] and
lemma[id[idx1]] != lemma[id[idx2]]:
                if str(id[idx1]) in tfidf:
                    tfidf1 = tfidf[str(id[idx1])]
                else:
                    tfidf1 = {}
                if str(id[idx2]) in tfidf:
                    tfidf2 = tfidf[str(id[idx2])]
                else:
                    tfidf2 = {}
                for iter1 in tfidf1:
                    if iter1 in tfidf2:
                        dotproduct += tfidf1[iter1]
            * tfidf2[iter1]
            if dotproduct != 0.0:
```

```

length1 = 0.0
for iter1 in tfidf1:
    length1 +=
math.pow(tfidf1[iter1], 2)
length1 = math.sqrt(length1)
length2 = 0.0
for iter2 in tfidf2:
    length2 +=
math.pow(tfidf2[iter2], 2)
length2 = math.sqrt(length2)
div = length1 * length2
result = dotproduct/div

```

Kode Sumber 4.14 Penghitungan Cosine Similarity

Implementasi penghitungan cosine similarity antar lemma menggunakan bahasa pemrograman python dan pustaka math untuk menghitung nilai kuadrat dan akar dari suatu angka. Fungsi `get_tfidf()` merupakan suatu fungsi yang mengembalikan nilai `tfidf` dari term hasil penghitungan sebelumnya. Pada fungsi cosine terdapat variabel data yang berisi id dari lemma dan lemma sehingga iterasi yang dilakukan adalah dilakukan setiap id dari lemma tersebut. Lalu hasil dari penghitungan cosine similarity tersebut akan diterapkan metode thresholding pada basis data dengan penghapusan kandidat sinonim dengan nilai threshold 0,7. Sehingga dihasilkan 67.544 sinonim pasang kata. Berikut beberapa data hasil pengolahan sinonim. Kode Sumber 4.13 dan Kode Sumber 4.14 merupakan potongan kode untuk penghitungan Cosine Similarity antar kata.

```

"lelaki": ["pameget", "mahram", "lanang", "doi",
           "bapak"],
"jihat": ["kanan", "belah", "kiri"],
"gabung": ["gebung", "bondot", "belembang", "barkas",
           "pocong"],
"maklumat": ["berita", "takrif", "pariwara"],
"plagiator": ["plagiat", "sistemisasi", "sinopsis",
              "nazam", "koral", "tinjau"],

```



```

"mikroorganisme": ["nyawa", "nematoda", "naluri",
                    "fisiologi", "enzim", "banzai", "organisme",
                    "semangat", "urip"],
"digit": ["eksponen", "angka", "jus", "anggota",
           "statistik"],
"jempol": ["jemari", "wulu cumbu", "telunjuk"],
"dagang": ["firma", "bilateralisme", "niagawan",
           "handelar", "saudagar"],
"korosif": ["kikis"],
"musim": ["titimangsa", "tempo", "sekarang", "kala"],

```

4.5 Ekspansi Pembentukan Graph Kerjasama dengan Sinonim Thesaurus.

Tahapan implementasi dari hasil proses perancangan Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus yang telah dilakukan pada Bab III. Seluruh tahapan implementasi Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus sesuai dengan perancangan pada Bab III menggunakan bahasa pemrograman python dan basis data SQL Server.

4.5.1 Data Peneliti dan Judul Publikasi

```

import pymssql
...
all_data = []
id_peneliti = []
judul = []

for item in id_peneliti:
    source_cursor.execute('SELECT judul FROM
tran_publicasi_dosen_tetap WHERE kode_pegawai = {}
ORDER BY tahun DESC, bulan DESC'.format(item))
    all_data = source_cursor.fetchall()
    title = ""
    for jud in all_data:
        if jud[0] is not None:
            title += " "+jud[0].encode('utf-8')
    if title != "":
        judul.append((item,title))

```

Kode Sumber 4.15 Inisiasi data publikasi peneliti

Kode Sumber 4.15 diatas merupakan kode implementasi dari pemrosesan awal judul peneliti untuk dikelompokkan per peneliti di ITS. Setelah dilakukan pengelompokan maka data judul penelitian disimpan ke dalam basis data untuk diterapkan praproses data dengan kode sumber yang sama dengan Tokenisasi, Penghapusan Tanda Baca Kalimat, Penghapusan Stop Word, Stemming Kata pada Bab IV Subbab Implementasi Pembentukan Sinonim Thesaurus Kateglo.

4.5.2 Modifikasi TF x IDF

E. Penghitungan Modifikasi TF

```
with open("sinonim.json", "r") as ins:
    for line in ins:
        lala = line
        sinonim = json.loads(lala.decode("utf-8"))
```

Kode Sumber 4.16 Mengambil Data Sinonim Dari File Json

```
...
c = {}
for item in judul:
    if item not in c.keys():
        c[item] = 0
    c[item] += 1
for item in c.keys():
    if item in sinonim:
        for sin in c.keys():
            if sin != item and sin in
sinonim[item]:
                c[item] += c[sin]
        norm = 1 + math.log10(float(c[item]))
...
```

Kode Sumber 4.17 Penghitungan Modifikasi Nilai TF

Implementasi penghitungan modifikasi TF dilakukan pertama kali adalah mengambil data sinonim kata pada file json sinonim yang telah digenerate pada Sub bab Implementasi Pembentukan Sinonim Thesaurus Kateglo. Variable judul

merupakan variable hasil pengolahan judul penelitian setelah dilakukan stemming kata. Pada perulangan pertama dilakukan penghitungan TF seperti algoritma TF biasa. Di perulangan yang kedua dilakukan pengecekan pada masing-masing hasil TF apakah dalam satu peneliti tersebut antar token term judul penelitiannya memiliki sinonim antar katanya. Setelah mendapatkan nilai TF dari sinonimnya maka tinggal dilakukan normalisasi nilai TF yang telah dimodifikasi. Setelah mendapatkan nilai TF yang telah dinormalisasi maka selanjutnya hasil TF dimasukkan ke basis data. Kode Sumber 4.16 dan Kode Sumber 4.17 merupakan potongan kode untuk implementasi penghitungan modifikasi TF.

F. Penghitungan Modifikasi IDF

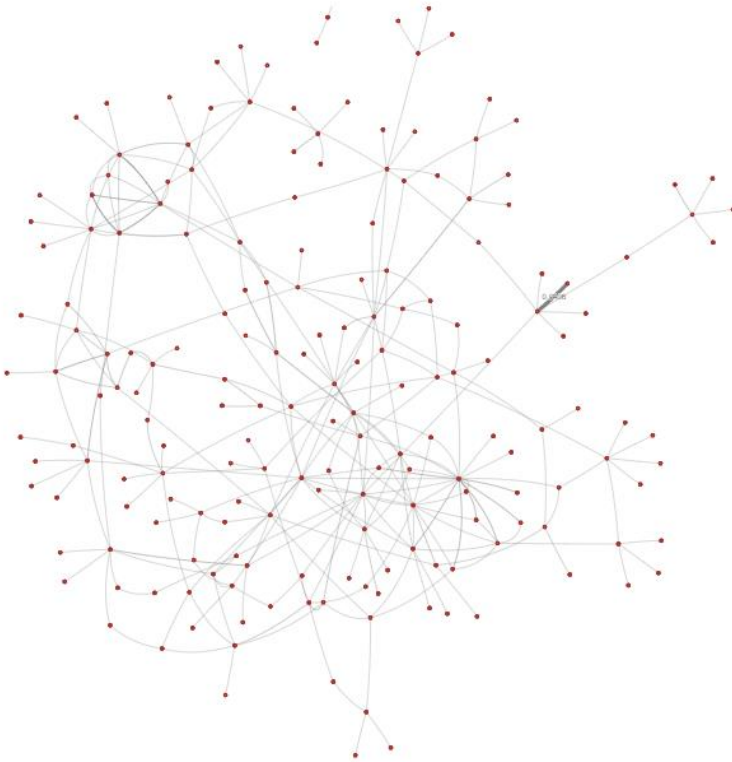
```
...
c = {}
for data in all_data:
    c[str(data[0])] = data[1]
for item in c.keys():
    if item in sinonim:
        for sin in sinonim[item]:
            if sin in c.keys():
                c[item] += c[sin]
    norm = math.log10( count / float(c[item]))
...
```

Kode Sumber 4.18 Penghitungan Modifikasi Nilai IDF

Implementasi penghitungan modifikasi IDF dilakukan pertama kali adalah mengambil data sinonim kata pada file json sinonim yang telah digenerate pada Sub bab Implementasi Pembentukan Sinonim Thesaurus Kateglo sama dengan pada penghitungan modifikasi nilai TF. Variable all_data merupakan variable hasil query pada basis data yang berisi struct(term,jumlah term muncul pada dokumen judul peneliti). Pada perulangan pertama dilakukan penghitungan IDF seperti algoritma IDF biasa. Di perulangan yang kedua dilakukan pengecekan pada masing-

masing hasil IDF apakah dalam term memiliki sinonim dalam term judul penelitian. Setelah mendapatkan nilai IDF dari sinonimnya maka tinggal dilakukan normalisasi nilai IDF yang telah dimodifikasi. Setelah mendapatkan nilai IDF yang telah dinormalisasi maka selanjutnya hasil IDF dimasukkan ke basis data. Setelah mendapatkan nilai IDF maka langkah selanjutnya adalah sama dengan

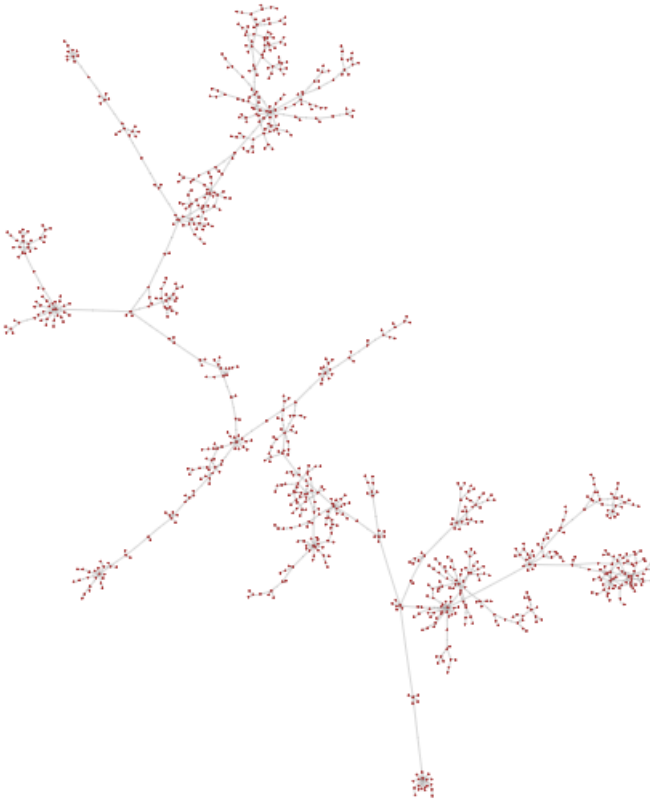
Penghitungan Bobot Kata TF IDF,



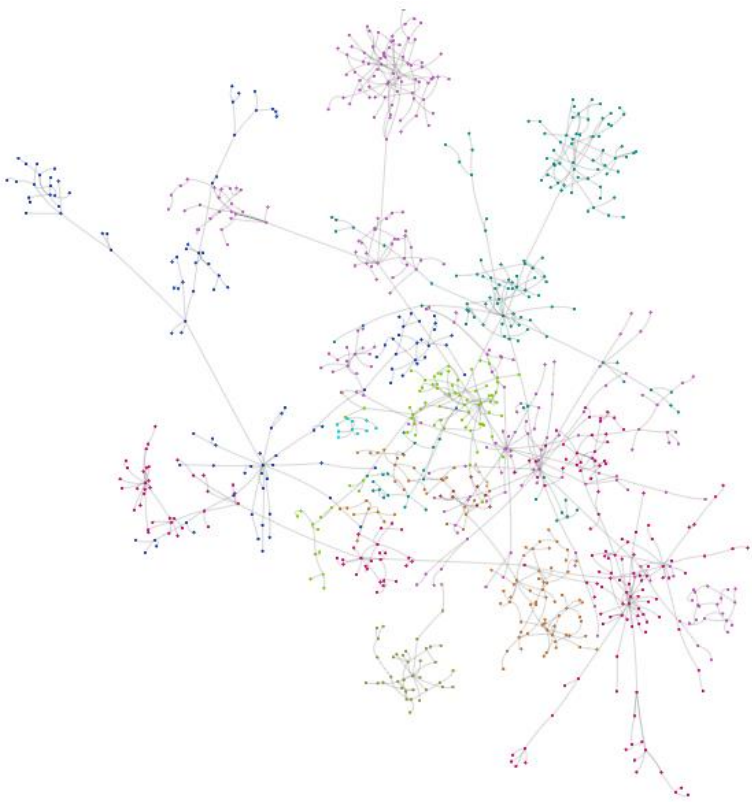
Gambar 4.6 Hasil Expand Pembentukan Graph Kerjasama dengan Sinonim Thesaurus

Penghitungan Cosine Similarity antar Judul Peneliti, Inisiasi Edge Value, Minimum Spanning Tree, Penerapan Prim Trajectory, dan Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster. Setelah menyelesaikan semua maka didapat graph kerjasama peneliti dengan memperhatikan sinonim kata pada judul penelitian sesuai dengan Gambar 4.6 – Gambar 4.8. Kode Sumber 4.18

merupakan potongan kode untuk penghitungan modifikasi IDF.



**Gambar 4.7 Hasil Minimum Spanning Tree dari Expand
Pembentukan Graph Kerjasama dengan Sinonim Thesaurus**



**Gambar 4.8 Hasil Clustering Expand Pembentukan Graph
Kerjasama dengan Sinonim Thesaurus**

BAB V

HASIL UJI COBA DAN EVALUASI

Bab ini berisi penjelasan mengenai skenario uji coba dan evaluasi pada tugas akhir. Tugas akhir ini akan memproses clustering peneliti berdasarkan topik peneliti menggunakan *k-means berbasis graph* dan *expand k-means berbasis graph dengan thesaurus yang dibentuk dari database katego*.

5.1 Deskripsi Uji Coba

Deskripsi uji coba menjelaskan lingkungan uji coba dan data uji coba yang digunakan untuk melakukan pengujian terhadap cluster kerjasama peneliti yang dibangun. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang digunakan, sedangkan data uji coba meliputi penjelasan tentang data yang digunakan pada uji coba.

5.1.1 Lingkungan Uji Coba

A. Perangkat Keras

Lingkungan implementasi perangkat keras dari tugas akhir ini adalah sebagai berikut:

- Tipe : PC Gygabyte Technology
- Prosesor : Inter® Core(TM) i5-3330 CPU (4 CPUs)
@ 3.2GHz
- Memori (RAM): 8192MB RAM

B. Perangkat Lunak

Lingkungan implementasi perangkat lunak dari tugas akhir ini adalah sebagai berikut:

- Sistem operasi : Windows 10 Enterprise 64-bit
- Web Server : Apache 2.4.4
- Basis Data : MySQL 5.5.32, SQL Server 2008

5.1.2 Data Uji Coba

Data yang digunakan untuk membangun dan melakukan uji coba pada tugas akhir ini adalah data judul penelitian dan data kerjasama antar peneliti di ITS. Data tersebut merupakan salinan dari data yang dimiliki oleh Pusat Data Terdistribusi ITS. Data kerjasama peneliti ITS memiliki 983 peneliti dan 25.261 data kerjasama keterkaitan antar dosen yang diambil dari basis data resits. Sedangkan untuk data judul penelitian terdapat 460.361 data.

5.1.3 Kuesioner Uji Coba

Kuesioner ujicoba perlu diterapkan pada tugas akhir ini karena graf kerjasama peneliti ITS tidak memiliki data tentang cluster kerjasama antar dosen, data cluster kerjasama diperlukan untuk menentukan ground truth dari pengujian kebenaran nilai cluster dari tugas akhir ini. Berikut target responden dan skenario pengisian kuesioner pada tugas akhir ini.

A. Target Responden

Target responden dari kuesioner ini adalah mahasiswa teknik informatika angkatan 2013 yang aktif dalam riset di ITS. Keaktifan mahasiswa dalam riset di ITS dapat diketahui melalui beberapa cara yaitu mahasiswa tersebut pernah terlibat atau tergabung dalam riset yang dimiliki oleh dosen, selanjutnya kriteria lainnya adalah mahasiswa tersebut aktif dalam laboratorium di Teknik Informatika sebagai administrator lab atau asisten laboratorium.

B. Skenario Pengisian Kuesioner

Skenario pengisian kuesioner dilakukan dalam dua tahap yaitu:

- a) Kuesioner tahap pertama memiliki tujuan mendapatkan nilai K terbaik dari hasil implementasi K-Means berbasis graf pada data Graf kerjasama peneliti di ITS yaitu K sama dengan 7 dapat dilihat pada Gambar 8.1 sampai Gambar 8.37, K sama dengan 10 dapat dilihat pada Gambar 8.38 sampai Gambar 8.70 dan K sama dengan 16 dapat dilihat pada Gambar 8.71 sampai Gambar 8.103. Setelah mendapatkan nilai K yang terbaik dan terburuk menurut responden, responden diminta untuk memberikan opini tentang apakah graf kerjasama yang telah diterapkan metode clustering lebih informatif dibandingkan sebelum diterapkan metode clustering. Hasil dari kuesioner tahap pertama akan digunakan untuk nilai K yang akan digunakan pada graf kerjasama peneliti ITS yang telah di ekspansi dengan sinonim. Responden juga diminta untuk menentukan dosen atau peneliti mana yang memiliki betweenness centrality tertinggi menurut responden.
- b) Kuesioner tahap kedua memiliki tujuan untuk mendapatkan graf mana yang lebih baik antara cluster graph kerjasama peneliti sebelum diekspansi dengan sinonim atau sesudah dengan nilai K yang sama. responden akan diminta opini untuk memilih graf mana yang menurut responden memiliki nilai kebenaran terbaik.

5.2 Uji Coba 1

5.2.1 Skenario Uji Coba 1

Skenario uji coba 1 bertujuan mendapatkan nilai K terbaik dari cluster yang terbentuk pada hasil Implementasi K-Means Berbasis Graph, dan mendapatkan 3 nilai K yaitu K 7, K 10, dan K 16 dari perubahan nilai threshold $C = 0.84$ dan $N = 15, 10, 7$ pada langkah Penerapan Prim Trajectory. Nilai K terbaik ditentukan

menggunakan metode pada pembahasan Cluster Analisis (*Dunn Index*).

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil penghitungan Cluster Analisis (*Dunn Index*) pada cluster kerjasama peneliti dari skenario uji coba 1 adalah sebagai berikut:

1. Tahap pertama yang dilakukan adalah memilih nilai K yang akan diproses terlebih dahulu yaitu data cluster peneliti dengan K sama dengan 7.
2. Tahap selanjutnya adalah melakukan penghitungan penghitungan Cluster Analisis (*Dunn Index*) pada data cluster kerjasama peneliti yang dipilih.
3. Selanjutnya catat semua hasil perhitungan yang didapan dari perhitungan diatas.
4. Setelah selesai kembali ke tahap 1 dan ganti data yang diproses dengan data cluster kerjasama peneliti dengan K dama dengan 10 dan selanjutnya 16.

5.2.2 Hasil Uji Coba 1

Tabel 5.1 Tabel Penghitungan Nilai Dunn Index dari Masing-masing Data Cluster Kerjasama Peneliti

K	Cluster	$\sum_{j=1}^{N_{C_i}} x_{i,j}$	N_{C_i}	$\delta(C_i)$
7	1	94.19571964	115	0.819093
	2	73.06938701	89	0.821004
	3	257.3729192	305	0.843846
	4	113.6315878	135	0.841715
	5	101.7524974	120	0.847937
	6	82.44423636	98	0.841268
	7	75.27618885	90	0.836402
	$\min_{1 \leq i \leq m} (\delta(C_i))$			0.819093

	$\max(\text{edge})$			0.9552
	$DI = \min_{1 \leq i \leq m} (\delta(C_i)) / \max(\text{edge})$			0.85751
10	1	94.19571964	115	0.819093
	2	52.78290502	64	0.824733
	3	19.41330513	24	0.808888
	4	118.037739	141	0.837147
	5	113.6315878	135	0.841715
	6	120.6892079	141	0.855952
	7	16.92784925	21	0.806088
	8	101.7524974	120	0.847937
	9	82.44423636	98	0.841268
	10	75.27618885	90	0.836402
	$\min_{1 \leq i \leq m} (\delta(C_i))$			0.806088
	$\max(\text{edge})$			0.9552
	$DI = \min_{1 \leq i \leq m} (\delta(C_i)) / \max(\text{edge})$			0.843895
16	1	75.81419332	93	0.815206
	2	52.78290502	64	0.824733
	3	19.41330513	24	0.808888
	4	99.5994835	118	0.844063
	5	10.68256348	13	0.821736
	6	17.58474599	22	0.799307
	7	83.33648427	99	0.841783
	8	29.43631941	35	0.841038
	9	6.186676225	8	0.773335
	10	120.6892079	141	0.855952
	11	16.92784925	21	0.806088
	12	101.7524974	120	0.847937
	13	17.51984217	21	0.834278

	14	63.85350744	75	0.85138
	15	18.81412926	23	0.818006
	16	55.61528909	66	0.842656
	$\min_{1 \leq i \leq m} (\delta(C_i))$			0.773335
	$\max(edge)$			0.9552
	$DI = \min_{1 \leq i \leq m} (\delta(C_i)) / \max(edge)$			0.809605

Sesuai dengan Tabel 5.1 diatas didapatkan nilai Dunn Index dari Cluster Kerjasama Peneliti Tertinggi terdapat pada nilai K sama dengan 7 dan nilai Dunn Index terendah terdapat pada cluster kerjasama peneliti dengan nilai K sama dengan 16.

5.2.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 1 didapatkan bahwa nilai K paling optimal adalah cluster kerjasama dengan K sama dengan 7 karena memiliki nilai Dunn Index paling kecil. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain itu adalah sebagai berikut:

1. Pada perhitungan dunn index sebuah graph akan memiliki nilai dunn index terbaik apabila nilai dari rata-rata antar cluster dalam graf memiliki nilai yang mirip/mendekati satu sama lain, karena rata-rata minimum nilainya tidak akan jauh beda dengan rata-rata maksimum.
2. Semakin banyak cluster yang terdapat pada suatu graph kemungkinan suatu graph memiliki nilai dunn index yang buruk semakin besar. Karena setiap data tersebar ke semua cluster dan nilai dunn index sangat tergantung pada keragaman rata-rata nilai edge per cluster. Sehingga semakin sedikit cluster yang

terbentuk, kemungkinan nilai dunn index semakin bagus pula.

3. Pada beberapa cluster terdapat tree yang terpisah meskipun masih dalam satu cluster, sebagai contoh Gambar 8.137 dan Gambar 8.139. dikarenakan tampilan graf filternya hanya menampilkan dosen FTIF maka vertex dosen non FTIF akan dihapus dari graf, oleh karena itu tree menjadi terpisah meskipun dalam satu cluster sebagai bukti terdapat pada Gambar 8.138 dan Gambar 8.140.

5.3 Uji Coba 2

5.3.1 Skenario Uji Coba 2

Skenario uji coba 2 bertujuan mendapatkan nilai K terbaik dari cluster yang terbentuk pada hasil Implementasi K-Means Berbasis Graph, dan mendapatkan 3 nilai K yaitu K 7, K 10, dan K 16 dari pengubahan nilai threshold pada langkah Penerapan Prim Trajectory. Nilai K terbaik ditentukan menggunakan metode kuesioner ke beberapa mahasiswa Teknik Informatika ITS. Uji coba dilakukan dengan menyebarkan kuesioner tahap 1 seperti pembahasan pada subbab 5.1.3. Data clustering kerjasama yang ditampilkan kepada responden adalah data kerjasama dosen peneliti ITS Fakultas Teknologi Informasi (FTIF) saja. Responden diminta untuk mengurutkan nilai K yang menurut responden paling mirip dengan kondisi actual penelitian di Fakultas Teknologi Informasi. Responden juga diminta memberikan pernyataan tentang alasan urutan jawaban terhadap kuesioner.

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan nilai K terbaik dengan menyebar kuesioner kepada mahasiswa Teknik Informatika angkatan 2013 adalah sebagai berikut:

1. Tampilkan semua graph yang akan diuji coba kepada responden

2. Berikan penjelasan secara singkat tentang bagaimana graph kerjasama terbentuk dan metode clustering yang dipakai.
3. Minta responden untuk menjawab pertanyaan nomor 1 yaitu mengurutkan cluster graph kerjasama dengan nilai K yang telah dihasilkan yang menurut responden memiliki nilai kebenaran paling mendekati dengan realitas penelitian di FTIF.
4. Minta responden untuk menjawab pertanyaan nomor 2 dan 3 tentang apakah cluster graph kerjasama yang dihasilkan lebih informatif daripada graph kerjasama yang belum diterapkan metode clustering.
5. Minta responden untuk menyebutkan alasan mengapa memilih jawaban tersebut.

5.3.2 Hasil Uji Coba 2

Tabel 5.2 Tabel Hasil Uji Coba 2

No	Item	Informasi
1	Nilai K yang menghasilkan cluster graf kerjasama terbaik	K7 = 1, K10 = 16, K16 = 15
2	Nilai K yang menghasilkan cluster graf kerjasama terjelek	K7 = 25, K10 = 0, K16 = 7
3	Nilai K terbaik lebih informatif dari graph kerjasama peneliti sebelum diterapkan metode clustering	Ya = 32, Tidak = 0
4	Nilai K terjelek lebih informatif dari graph kerjasama peneliti sebelum diterapkan metode clustering	Ya = 28, Tidak = 4
5	Total jumlah responden	32
6	Jumlah responden yang aktif dalam penelitian di ITS	29

Sesuai dengan tabel 5.2 didapatkan bahwa cluster dengan nilai K terbaik adalah cluster kerjasama peneliti dengan nilai K sama dengan 10.

5.3.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 2 bahwa cluster dengan nilai K terbaik adalah cluster kerjasama peneliti dengan nilai K sama dengan 10. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain informasi tersebut adalah sebagai berikut:

1. Graph yang telah dicluster / atau diterapkan metode K-Means berbasis graf lebih informatif daripada graf kerjasama peneliti yang belum diterapkan metode clustering. Karena setiap cluster graph yang terbentuk memiliki informasi terkait edge dengan topik keterkaitan antar peneliti dan memiliki informasi lain berupa kelompok – kelompok yang merupakan representasi dari fokus penelitian.
2. Sebagian besar beralasan bahwa nilai K sama dengan 10 adalah yang terbaik karena graph dengan K sama dengan 7 terlalu besar dalam mengelompokkan peneliti di FTIF dan K sama dengan 16 terlalu kecil memetakan kerjasama antar peneliti hingga ada cluster yang terbentuk dari 2 vertex saja.
3. Pada Gambar 8.141 merupakan jurusan yang ada di ITS dan sebagian besar topik penelitian berfokus pada teknologi informasi. Garis penghubung antar jurusan pada gambar tersebut merepresentasikan jumlah cluster yang sama dalam satu jurusan. Cluster yang sama dalam satu jurusan, mengindikasikan bahwa antar jurusan yang terhubung memiliki keterkaitan topik penelitian.

5.4 Uji Coba 3

5.4.1 Skenario Uji Coba 3

Skenario uji coba 3 bertujuan membandingkan nilai dunn index dari Nilai K terbaik hasil dari Uji Coba 1 yaitu $K = 16$ dan graf Ekspansi Pembentukan Graph Kerjasama dengan Sinonim Thesaurus. Nilai K yang dipakai pada graph kerjasama dengan ekspansi sinonim kata disamakan dengan nilai K sama dengan 16 untuk mendapatkan hasil yang seimbang antar cluster. Uji coba dilakukan sebanyak dua kali dengan menggunakan graf cluster kerjasama peneliti dengan K 16, dan graf kerjasama dengan ekspansi sinonim kata dengan K sama dengan 16.

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil graph mana yang lebih baik dari skenario uji coba 3 adalah sebagai berikut:

1. Tahap pertama yang dilakukan adalah memilih nilai K yang akan diproses terlebih dahulu yaitu data cluster peneliti dengan K sama dengan 16.
2. Tahap selanjutnya adalah melakukan penghitungan penghitungan Cluster Analisis (*Dunn Index*) pada data cluster kerjasama peneliti yang dipilih.
3. Selanjutnya catat semua hasil perhitungan yang didapan dari perhitungan diatas.
4. Setelah selesai kembali ke tahap 1 dan ganti data yang diproses dengan data graf cluster kerjasama peneliti ekspansi sinonim kata dengan K sama dengan 16.

5.4.2 Hasil Uji Coba 3

Tabel 5.3 Tabel Hasil Uji Coba 3

Jenis	Cluster	$\sum_{j=1}^{N_{C_i}} x_{i,j}$	N_{C_i}	$\delta(C_i)$
Cluster	1	75.81419332	93	0.815206
	2	52.78290502	64	0.824733
	3	19.41330513	24	0.808888

	4	99.5994835	118	0.844063
	5	10.68256348	13	0.821736
	6	17.58474599	22	0.799307
	7	83.33648427	99	0.841783
	8	29.43631941	35	0.841038
	9	6.186676225	8	0.773335
	10	120.6892079	141	0.855952
	11	16.92784925	21	0.806088
	12	101.7524974	120	0.847937
	13	17.51984217	21	0.834278
	14	63.85350744	75	0.85138
	15	18.81412926	23	0.818006
	16	55.61528909	66	0.842656
	$\min_{1 \leq i \leq m} (\delta(C_i))$			0.773335
	$\max(edge)$			0.9552
	DI			0.809605
Cluster graph kerjasama peneliti expand sinonim kata	1	287.5447078	709	0.405564
	2	3.478356391	9	0.386484
	3	1.395183295	4	0.348796
	4	2.299767137	7	0.328538
	5	20.0357576	48	0.417412
	6	2.95237726	8	0.369047
	7	3.121999413	8	0.39025
	8	0.649460465	2	0.32473
	9	3.120188504	9	0.346688
	10	0.982168436	3	0.327389
	11	10.27012792	24	0.427922
	12	7.600701541	18	0.422261
	13	4.821291745	12	0.401774

	14	4.722211331	12	0.393518
	15	4.748349398	12	0.395696
	16	4.338451475	10	0.433845
	$\min_{1 \leq i \leq m} (\delta(C_i))$			0.32473
	$\max(\text{edge})$			0.497719
	DI			0.652436

Sesuai dengan tabel 5.3 diatas didapatkan nilai Dunn Index dari cluster kerjasama peneliti ekspansi sinonim kata lebih buruk daripada cluster kerjasama peneliti tanpa ekspansi sinonim kata. Karena semakin besar nilai Dunn Index suatu cluster graf maka semakin bagus pula nilai Dunn Index dari suatu graf tersebut.

5.4.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 3 didapatkan nilai Dunn Index dari cluster kerjasama peneliti ekspansi sinonim kata lebih buruk daripada cluster kerjasama peneliti tanpa ekspansi sinonim kata. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain hal tersebut adalah sebagai berikut:

1. Graf cluster kerjasama peneliti ekspansi sinonim kata memiliki rentang nilai antara 0 - 0.497719 dan Cluster graf kerjasama peneliti memiliki rentang nilai edge 0 - 0.9552, dikarenakan nilai edge yang dihasilkan lebih homogen antar kata setelah diterapkan sinonim kata. Nilai yang homogen pada uji cluster graph kerjasama peneliti ekspansi sinonim kata memiliki dikarenakan keunikan judul antar peneliti berkurang. Hal tersebut berdampak pada terbentuknya super cluster dengan index 1 dengan anggota cluster 709 peneliti.
2. Dalam perspektif nilai dunn index cluster graph kerjasama peneliti expand sinonim kata dengan K

sama dengan 16 memiliki nilai yang buruk yaitu 0.652436 karena dalam hal persebaran peneliti nilai K sama dengan 16 sangat buruk, karena jumlah anggota cluster yang tidak seimbang. Hal ini disebabkan karena nilai pada graf cluster kerjasama peneliti ekspansi sinonim kata memiliki nilai yang homogen sesuai dengan penjelasan pada poin 1.

5.5 Uji Coba 4

5.5.1 Skenario Uji Coba 4

Skenario uji coba 4 adalah skenario uji coba dengan membandingkan hasil dari Uji Coba 2 yaitu $K = 10$ dan graf Ekspansi Pembentukan Graph Kerjasama dengan Sinonim Thesaurus. Nilai K yang dipakai pada graph kerjasama dengan ekspansi sinonim kata disamakan dengan nilai K sama dengan 10. Cluster graph yang lebih baik ditentukan menggunakan metode kuesioner sesuai dengan subbab 5.1.3. Uji coba dilakukan dengan menyebarkan kuesioner terhadap mahasiswa Teknik informatika ITS angkatan 2013. Data clustering kerjasama yang ditampilkan kepada responden adalah data kerjasama dosen peneliti ITS Fakultas Teknologi Informasi (FTIF) saja. Responden diminta untuk memilih antara cluster graph kerjasama peneliti dan cluster graph kerjasama peneliti expand sinonim kata yang paling mirip dengan kondisi aktual penelitian di Fakultas Teknologi Informasi. Responden juga diminta memberikan pernyataan tentang alasan urutan jawaban terhadap kuesioner.

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil penghitungan waktu proses dari skenario uji coba 1 adalah sebagai berikut:

1. Tampilkan semua graph yang akan diuji coba kepada responden
2. Berikan penjelasan secara singkat tentang bagaimana graph kerjasama terbentuk dan metode clustering yang dipakai dan pembentukan sinonim kata kataglo.

3. Minta responden untuk menjawab pertanyaan nomor 1 yaitu memilih antara cluster graph kerjasama peneliti dan cluster graph kerjasama peneliti expand sinonim kata yang paling mirip dengan kondisi aktual penelitian di Fakultas Teknologi Informasi..
4. Minta responden untuk menyebutkan alasan mengapa memilih jawaban tersebut.

5.5.2 Hasil Uji Coba 4

Tabel 5.4 Tabel Hasil Ujicoba 4

No	Item	Informasi
1	Hasil cluster yang lebih baik	Graf cluster kerjasama peneliti ITS = 21 Graf cluster kerjasama peneliti ITS ekspansi sinonim kata = 0
2	Graf cluster kerjasama peneliti ekspansi sinonim kata lebih informatif dari graf kerjasama peneliti ITS sebelum diterapkan metode clustering	Ya = 18, Tidak = 3
3	Total jumlah responden	21
4	Jumlah responden yang aktif dalam penelitian di ITS	20

Tabel 5.4 menunjukkan hasil bahwa graf cluster kerjasama peneliti tanpa ekspansi sinonim kata lebih baik daripada graf cluster kerjasama peneliti ekspansi sinonim kata menurut seluruh responden.

5.5.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 4 didapatkan hasil Clustering graph kerjasama peneliti ITS dipilih oleh seluruh responder lebih sesuai dengan realita daripada cluster kerjasama peneliti expand sinonim kata.. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain informasi tersebut adalah sebagai berikut:

1. Penggunaan sinonim dalam penghitungan TF IDF nampaknya semakin membuat karakteristik kata penelitian semakin hilang, sehingga graph yang dihasilkan tidak sesuai dengan kerjasama peneliti pada kondisi nyata menurut responder.
2. Sinonim yang dihasilkan dari pengolahan definisi kamus kata kateglo tidak begitu optimal, karena definisi antar kata yang memiliki sinonim terkadang tidak sama sehingga bobot kemiripan kata menjadi rendah. Seperti contoh sinonim yang dihasilkan dibawah ini:

```
"bengku": ["fotogravur", "energi",
"disinfektan", "cocok tanam", "cekok",
"cangkul", "bujet", "biotron", "belacan",
"barometer"],
"hegemoni": ["dominion", "warga negara",
"upeti", "tanah", "reekspor", "persona
nongrata", "misi", "konsul", "konkordat",
"ekstradisi"],
"koma": ["kemukus"],
"cecunguk": ["kecoak"],
"wacana": ["istitaah", "daya", "kinerja"],
"desil": ["kuartil", "kuintil",
"presentil"],
```

3. Secara garis besar ekspansi sinonim kata pada pembentukan basis data kateglo tidak begitu berdampak pada letak vertex dalam suatu graf. Sebagai bukti pada Tabel 8.1 dan Tabel 8.2 terdapat daftar dosen yang memiliki sinonim kata paling sedikit dalam termnya. Daftar dosen yang tercantum dalam daftar tersebut juga tidak konsisten nilai betweenness nya karena ada satu dosen yang memiliki betweenness tinggi meskipun termasuk dalam daftar tersebut.

5.6 Uji Coba 5

5.6.1 Skenario Uji Coba 5

Dalam *social network analysis* terdapat suatu vertex yang dianggap sebagai pusat suatu cluster atau graf karena tepat berada ditengah suatu graf. Pada skenario uji coba 2 graf kerjasama peneliti ITS FTIF terdapat kelas terbesar. Dari cluster tersebut dicari vertex yang paling central posisinya menggunakan metode betweenness pada bagian 2.5 Betweenness Centrality.

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil central vertex dari skenario uji coba 5 adalah sebagai berikut:

1. Load semua data yang dimiliki cluster tersebut.
2. Hitung nilai betweenness vertex pada cluster tersebut.
3. Setelah selesai kembali ke langkah 2 dan hitung nilai betweenness dari semua vertex graph tersebut

Dari Gambar 5.1 **Rangking Nilai Betweenness Peneliti pada cluster terbesar FTIF** peneliti dengan nilai betweenness adalah ibu Henning Titi Ciptaningtyas dengan nilai 1631. Pada Gambar 5.2 **Visualisasi Node Peneliti berdasarkan nilai Betweenness** peneliti yang memiliki warna hijau bisa dikategorikan sebagai vertex yang berada di tepi graph. Terdapat 6 peneliti yang memiliki nilai betweenness tinggi.

5.6.2 Analisis dan Evaluasi

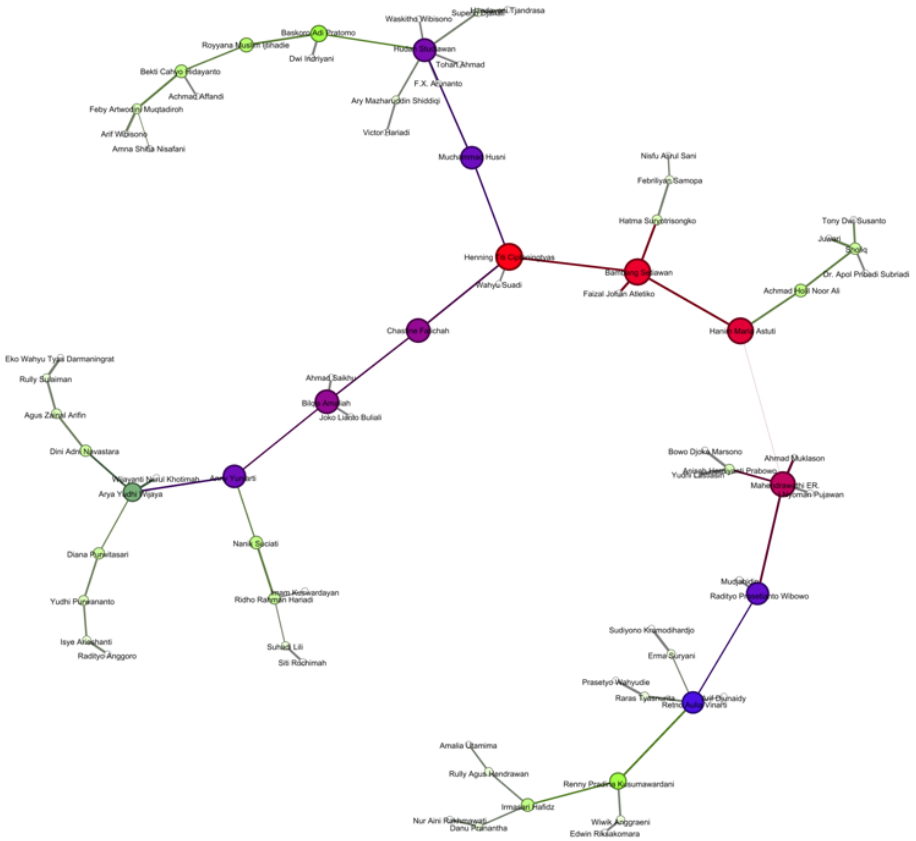
Pada hasil skenario uji coba 5 didapatkan hasil peneliti dengan nilai betweenness adalah ibu Henning Titi Ciptaningtyas dengan nilai 1631. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain informasi tersebut adalah sebagai berikut:

1. Terdapat 6 peneliti yang memiliki nilai betweenness tinggi yaitu Hening Titi Ciptaningtyas, Bambang Setiawan, Hanim Maria Astuti, Mahendrawathi ER, Chastine Fatichah, Bilqis Amaliah.

5.6.3 Hasil Uji Coba 5

Rank	Label
1631.0	Henning Titi Ciptaningtyas
1363.0	Bambang Setiawan
1298.0	Hanim Maria Astuti
1137.0	Mahendrawathi ER.
988.0	Chastine Fatichah
987.0	Bilqis Amaliah
915.0	Hudan Studiawan
890.0	Anny Yuniarti
880.0	Muchammad Husni
854.0	Radityo Prasetyanto Wib...
802.0	Retno Aulia Vinarti
582.0	Arya Yudhi Wijaya
458.0	Renny Pradina Kusuma...
454.0	Baskoro Adi Pratomo
330.0	Royyana Muslim Ijtihadie
272.0	Irmasari Hafidz
271.0	Bekti Cahyo Hidayanto
268.0	Nanik Suciati
268.0	Achmad Holil Noor Ali
207.0	Sholih
206.0	Ridho Rahman Hariadi
204.0	Diana Purwitasari
204.0	Dini Adni Navastara
139.0	Feby Artwodini Muqtadi...
139.0	Anisah Herdiyanti Prabo...
138.0	Yudhi Purwananto
138.0	Hatma Suryotrisongko
138.0	Agus Zainal Arifin

Gambar 5.1 Rangking Nilai Betweenness Peneliti pada cluster terbesar FTIF



Gambar 5.2 Visualisasi Node Peneliti berdasarkan nilai Betweenness

2. Topik yang dimiliki oleh vertex tertinggi dari cluster terbesar dalam graf kerjasama peneliti FTIF ITS adalah development, high, main, result, optimal, performance, gas, algorithm, integrated, level, number, algoritma, network, multi, systems, problems, rate, motor, transportasi, penilaian, jaringan, approach, case, quality, sensor, increase, application, oil, software, production, java, increasing, citra, optimasi, average, pengendalian, error, jarak, point, generator, penanggulangan, vehicle, addition, developed, road, experimental, volume, implemented, monitoring, game, improve, controller, series, activities, fuel, conditions, processing, kendaraan, strategis, real, standar, response, years, graph, route, neural, technique, respon, current, reduce, basis, safety, cycle, learning, caused, lunak, fluid, mobile, batas, web, modul, medium, source, change, dirancang, fluida, stations, provide, digital, service, angle, dinamis, keamanan, arah, highest, existing, people, pariwisata, open, bali, position, kualitatif, accuracy, pelatihan, fitur, perkotaan, experiment, event, testing, performa, artificial, generated, beda, purpose, prediction, objek, detection, aims, akurasi, types, productivity, peta, tracking, adaptive, general, developing, kelas, suitable, spasial, terminal, center, applications, morphology, framework, automatic, video, infrastructure, critical, operating, prediksi, mikrokontroler.
3. Topik yang dimiliki oleh vertex yang memiliki nilai betweenness tertinggi bisa menjadi representasi topik suatu cluster karena vertex tersebut berada tepat ditengah suatu cluster yang bobot antar edge dilakukan penghitungan dari topik penelitian. Pada Tabel 8.3 terdapat daftar term topik yang mewakili setiap dosen FTIF ITS.

5.7 Uji Coba 6

5.7.1 Skenario Uji Coba 6

Dalam social network analysis terdapat suatu vertex yang dianggap sebagai pusat suatu cluster atau graph karena tepat berada ditengah suatu graph. Pada skenario uji coba 2 dari graph kerjasama peneliti FTIF ITS terdapat kelas terbesar. Dari cluster tersebut dicari vertex yang paling central posisinya menggunakan kuesioner ke beberapa mahasiswa Teknik Informatika ITS. Uji coba dilakukan dengan menyebarkan kuesioner terhadap mahasiswa Teknik informatika ITS angkatan 2013. Pengambilan data kuesioner dilakukan bersamaan dengan pengambilan pada Uji Coba .

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil central vertex dari skenario uji coba 6 adalah sebagai berikut:

1. Tampilkan data cluster kerjasama khususnya cluster terbesar yang ada pada cluster kerjasama peneliti FTIF ITS.
2. Jelaskan kepada responden tentang cara menghitung dan pengertian dari nilai betweenness centrality.
3. Minta responden untuk memilih 1 peneliti dari data tersebut yang menurut responden memiliki nilai betweenness tertinggi.
4. Ulangi langkah 1 – 3 kepada beberapa mahasiswa Teknik Informatika ITS.

5.7.2 Hasil Uji Coba 6

Tabel 5.5 Tabel Hasil Uji Coba 6

No	Peneliti	Jumlah pilihan
1	Hening Titi Ciptaningtyas	20
2	Hudan Studiawan	9
3	Chastine Fatichah	2

4	Hanim Maria Astuti	1
5	Total jumlah responden	32
6	Jumlah responden yang aktif dalam penelitian di ITS	29

Pada hasil penarikan data melalui kuesioner pada Tabel 5.5 didapatkan bahwa ibu Hening Titi Ciptaningtyas menjadi peneliti dengan nilai betweenness tertinggi menurut para responden. Keaktifan responden dalam penelitian di ITS dinilai dari beberapa hal yaitu pernah mengikuti riset yang dilakukan oleh dosen, pernah tergabung ke dalam project yang ditangani dosen, atau menjadi admin salah satu lab di Teknik Informatika.

5.7.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 6 didapatkan hasil berupa dosen yang memiliki nilai betweenness tertinggi adalah ibu Hening Titi Ciptaningtyas. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain informasi tersebut adalah sebagai berikut:

1. Responder yang memilih ibu Henning Titi Ciptaningtyas dan ibu Chastine Fatichah sebagai peneliti dengan nilai betweenness tertinggi karena peneliti tersebut tepat berada di tengah cluster tersebut.
2. Berbeda dengan responder yang memilih Bapak Hudan Studiawan sebagai vertex dengan nilai betweenness tertinggi, responder memiliki alasan apabila peneliti tersebut dihapus dari cluster graf tersebut, responder beranggapan graf tersebut tidak akan memiliki informasi yang sesuai lagi seperti semula. Hal ini disebabkan bapak Hudan Studiawan memiliki cukup banyak edge yang terhubung dengan banyak vertex lain meskipun lokasinya tidak tepat berada ditengah cluster.
3. Selain itu ada beberapa responder yang menentukan ibu Hanis Maria Astuti sebagai vertex dengan nilai

betweenness tertinggi dengan alasan edge yang dimiliki ibu Hanim memiliki nilai edge terkecil dari graph tersebut. Responden beranggapan bahwa waktu pembentukan minimum spanning tree bu hanim terpilih sebagai initial vertex karena memiliki edge terkecil.

5.8 Uji Coba 7

5.8.1 Skenarioa Uji Coba 7

Pada graph kerjasama peneliti ITS dengan nilai N sama dengan 7, 10 atau 16 selalu menciptakan cluster besar yang dominan dalam hasil clusteringnya. Skenario uji coba 7 bertujuan untuk mendapatkan nilai variabel pada Penerapan Prim Trajectory agar cluster besar tersebut terbagi menjadi beberapa cluster. Nilai variabel C dan N (C,N) pada penerapan prim trajectory yang digunakan uji coba kali ini adalah (0.55,1), (0.55,2), (0.6,1), (0.6,2), (0.7,1), (0.7,2).

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil penghitungan waktu proses dari skenario uji coba 7 adalah sebagai berikut:

1. Ubah nilai C dan N pada implementasi prim trajectory dengan nilai (0.55,1)..
2. Jalangkan kembali implementasi Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster.
3. Periksa apakah cluster besar tersebut sudah terpecah atau belum.
4. Ulangi langkah 1 sampai 3 dengan nilai C dan N sama dengan nilai (0.55,2), (0.6,1), (0.6,2), (0.7,1), (0.7,2).

5.8.2 Hasil Uji Coba 7

Tabel 5.6 Tabel Hasil Uji Coba 7

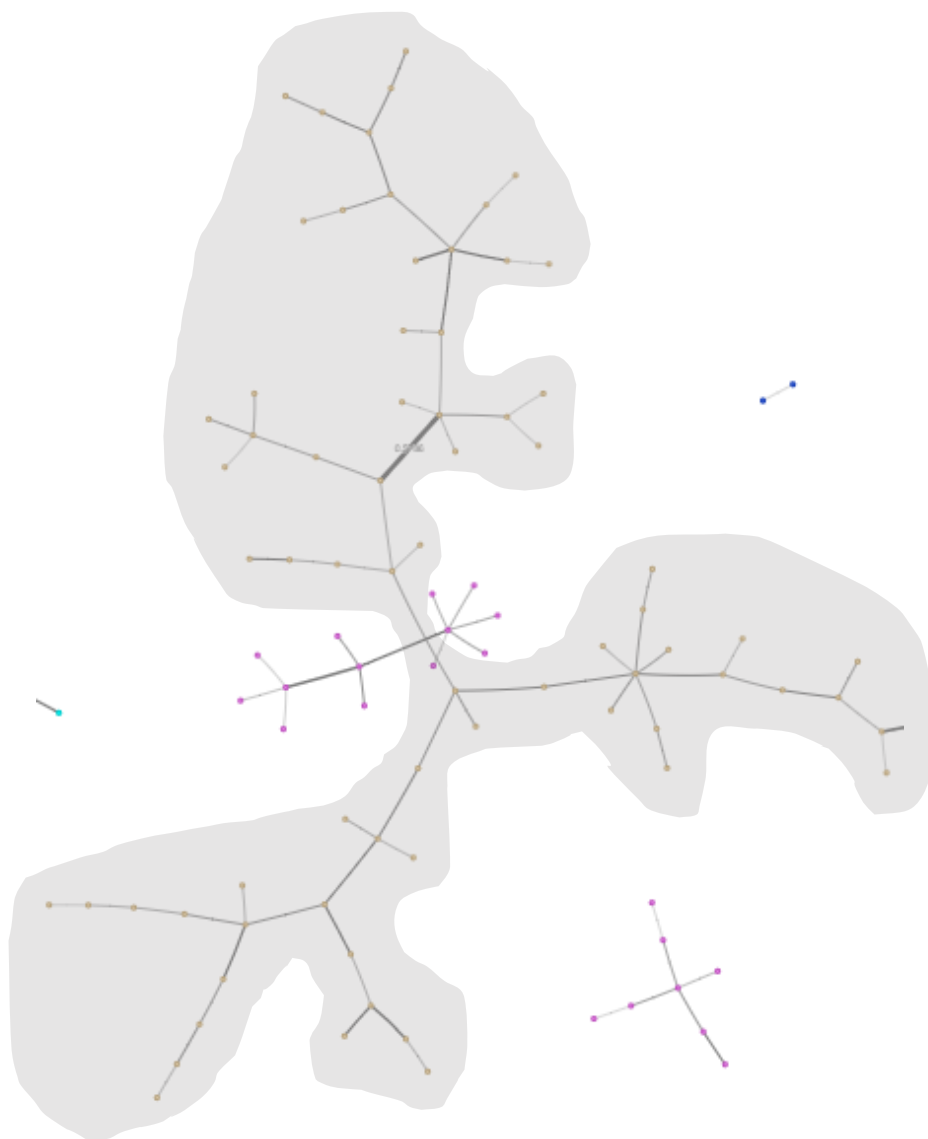
Nilai C	Nilai K	Terjadi perubahan pada cluster besar
---------	---------	--------------------------------------

0.55	1	Tidak
0.55	2	Tidak
0.6	1	Ya
0.6	2	Tidak
0.7	1	Tidak
0.7	2	Tidak

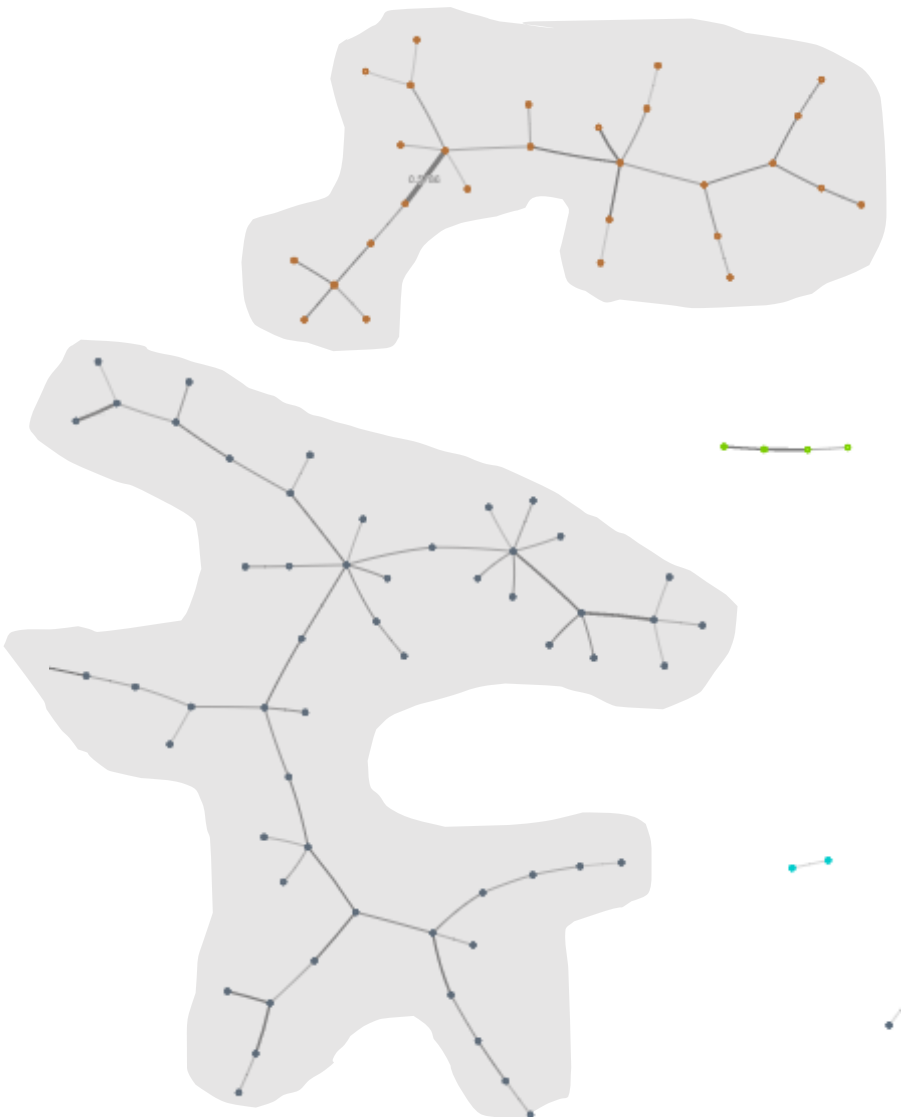
5.8.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 7 pada Tabel 5.6 didapatkan hasil bahwa nilai C dan N yang menghasilkan graph kerjasama yang berbeda pada peneliti di FTIF adalah dengan nilai (0.6,1). Beberapa hal yang dapat diambil dari hasil uji coba di atas selain informasi tersebut adalah sebagai berikut:

1. Untuk mendapatkan nilai C yang optimal perlu memperhitungkan jumlah data yang didapat dari hasil thresholding pada nilai edge, semakin besar data yang dikembalikan dari hasil thresholding maka semakin banyak nilai N yang dapat dikombinasikan agar cluster graph kerjasama dapat tercipta, dan sebaliknya. Misal diberikan nilai C dan N sama dengan (0.5,5) cluster graph kerjasama tidak akan terbentuk karena jumlah edge yang nilai jaraknya kurang dari 0.5 hanya 10 buah edge, dan setelah dilakukan filtering sub cluster dengan nilai N sangat kecil kemungkinan sub cluster tersebut lolos dari threshold nilai N.
2. Nilai N yang optimal adalah dengan memperhitungkan dari jumlah data edge yang didapat hasil thresholding dengan nilai C. Apabila jumlah edge yang dihasilkan kurang dari sepertiga jumlah edge, maka nilai N yang dianjurkan untuk kasus graph kerjasama peneliti ITS adalah nilai N kurang dari 3.



Gambar 5.3 Cluster Besar pada Graph Kerjasama Peneliti FTIF



Gambar 5.4 Hasil Pemecahan Cluster Besar pada Graph Kerjasama Peneliti FTIF

5.9 Uji Coba 8

5.9.1 Skenario Uji Coba 8

Skenario uji coba 8 adalah skenario penghitungan waktu eksekusi program yang dibutuhkan untuk membangun cluster graph kerjasama antar peneliti di ITS. Skenario ini bertujuan untuk mengetahui jumlah waktu yang dibutuhkan untuk membangun clustering graph kerjasama peneliti ITS berdasarkan data uji coba.

Uji coba dilakukan dan dicatat selama pembentukan clustering graph kerjasama peneliti ITS. Proses yang dimaksud adalah semua proses yang terdapat pada

Tahapan-tahapan uji coba yang dilakukan untuk mendapatkan hasil penghitungan waktu proses dari skenario uji coba 8 adalah sebagai berikut:

1. Jalankan setiap langkah program dan catat waktu mulai dari program.
2. Ketika program selesai berjalan, catat waktu ketika ketika program berjalan.

5.9.2 Hasil Uji Coba 8

Tabel 5.7 Tabel Waktu Eksekusi

Proses	Sub Proses	Waktu (menit)
K-Means Berbasis Graph Data Kerjasama Peneliti ITS	a. Inisiasi Edge Value	9
	b. Pembentukan Minimum Spanning Tree	14
	c. Penerapan Prim Trajectory	4
	d. Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	7
Pembentukan Thesaurus	a. Praproses Data	12
	b. Penghitungan TF x IDF	231

Sinonim dari Kamus Kata Kateglo	c. Penghitungan Cosine Similarity antar Kata	5632
	d. Filter Kandidat Sinonim Kata	31
Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus Sinonim	a. Praproses Data	12
	b. Modifikasi penghitungan TF	8
	c. Modifikasi penghitungan IDF	6
	d. Penghitungan TF x IDF	12
	e. Inisiasi Edge Value	8
	f. Pembentukan Minimum Spanning Tree	11
	g. Penerapan Prim Trajectory	5
	h. Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	6

Pembentukan cluster graf dengan metode K-Means berbasis graph pada Tabel 5.7 dan Tabel 5.8 memiliki rata-rata waktu eksekusi kurang dari 10 menit. Sedangkan untuk pembentukan thesaurus sinonim membutuhkan hampir 4 hari pada bagian penghitungan cosine similarity. Untuk rata-rata waktu pembentukan graph kerjasama dengan thesaurus sinonim juga memiliki rata-rata eksekusi kurang dari 10 menit.

Tabel 5.8 Tabel Jumlah Data yang Dihasilkan

Proses	Sub Proses	Jumlah Data
K-Means Berbasis Graph Data Kerjasama Peneliti ITS	a. Inisiasi Edge Value	460.361 record kerjasama antar dosen
	b. Pembentukan Minimum Spanning Tree	958 edge
	c. Penerapan Prim Trajectory	958 edge
	d. Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	958 edge

Pembentukan Thesaurus Sinonim dari Kamus Kata Kateglo	a. Praproses Data	243.512 dokumen lemma
	b. Penghitungan TF x IDF	33.189 dokumen lemma
	c. Penghitungan Cosine Similarity antar Kata	14.082.888 bobot sinonim lemma
	d. Filter Kandidat Sinonim Kata	67.544 bobot sinonim kata
Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus Sinonim	a. Praproses Data	25.261 judul penelitian
	b. Modifikasi penghitungan TF	61.979 kata dalam judul penelitian
	c. Modifikasi penghitungan IDF	18.573 kata dalam judul penelitian
	d. Penghitungan TF x IDF	61.979 kata dalam judul penelitian
	e. Inisiasi Edge Value	439.516 record kerjasama antar dosen
	f. Pembentukan Minimum Spanning Tree	958 edge
	g. Penerapan Prim Trajectory	958 edge
	h. Penghitungan Cluster pada Vertex yang Belum Memiliki Cluster	958 edge

5.9.3 Analisis dan Evaluasi

Pada hasil skenario uji coba 8 didapatkan hasil waktu yang *linear* atau berbanding lurus, yaitu semakin banyak data yang diproses, maka semakin lama pula waktu yang dibutuhkan untuk menjalankan program. Beberapa hal yang dapat diambil dari hasil uji coba di atas selain waktu yang dibutuhkan adalah sebagai berikut:

1. Rata-rata waktu eksekusi untuk metode K-Means berbasis graph memiliki waktu kurang dari 10 menit. Rata-rata waktu eksekusi untuk pembentukan sinonim adalah 1476.5 menit atau kurang lebih sama dengan 24 jam. Rata-rata waktu eksekusi Ekspansi Pembentukan Graph Kerjasama dengan Thesaurus sinonim memiliki waktu kurang dari 10 menit.
2. Proses yang memiliki waktu paling lama adalah Penghitungan Cosine Similarity antar Kata pada pembentukan thesaurus sinonim dari kamus kata kateglo. Proses tersebut memiliki waktu yang sangat lama karena memiliki kompleksitas algoritma n^2 . Setiap lemma harus dipasangkan dengan lemma yang lain untuk mendapatkan bobot kedekatan antar lemma. Jumlah masukan data pada proses ini adalah 33.189 data dokumen lemma dan memiliki keluaran data sebanyak 14.082.888 data bobot sinonim antar lemma. Data yang dihasilkan pada proses perhitungan cosine similarity antar kata pada pembentukan thesaurus sinonim dari kamus kata kateglo tidak semua lemma memiliki pasangan kepada setiap lemma. Karena apabila nilai cosine yang menghasilkan nilai 0 tidak di teruskan prosesnya.
3. Proses yang memiliki waktu paling singkat adalah Penerapan Prim Trajectory yaitu 4 menit dikarenakan dalam proses ini yang dilakukan adalah thresholding data dan data yang diolah hanya data edge hasil dari proses pembentukan minimum spanning tree yang telah memenuhi threshold.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

6.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil uji coba pembentukan cluster kerjasama peneliti ITS menggunakan metode K-Means berbasis graph adalah sebagai berikut:

1. Nilai K terbaik menurut penghitungan dunn index terhadap cluster kerjasama peneliti adalah K sama dengan 16. Semakin banyak cluster yang terdapat pada suatu graph kemungkinan suatu graph memiliki nilai dunn index yang baik semakin besar. Karena setiap data tersebar ke semua cluster dan nilai dunn index sangat tergantung pada rata-rata nilai edge per cluster. Sehingga semakin banyak cluster kemungkinan nilai dunn index semakin bagus pula.
2. Berdasarkan hasil ujicoba 3, graph yang telah dicluster lebih informatif daripada graph yang belum dicluster. Karena clusterisasi pada graph peneliti ITS memunculkan informasi topik-topik yang dominan dalam suatu cluster.
3. Penerapan ekspansi sinonim kata pada penghitungan TF dan IDF memiliki dampak pada nilai kemiripan antar topik peneliti, nilai yang dihasilkan memiliki simpangan nilai yang lebih kecil daripada pembentukan graph kerjasama yang menggunakan penghitungan TF dan IDF biasa.
4. Sinonim kata yang dihasilkan dari pembentukan sinonim kata dari kamus data kateglo masih belum optimal karena definisi kata dari basis data kateglo masih belum berminkin bahwa kata yang bersinonim memiliki definisi yang sama.

6.2 Saran

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini adalah:

1. Pembentukan sinonim kata menggunakan kamus data kateglo tidak begitu optimal karena definisi antar kata yang masih belum akurat. Oleh karena itu beberapa website juga menyediakan pencarian sinonim kata seperti sinonimkata.com, untuk meningkatkan kualitas definisi kata bisa menggunakan penggabungan definisi kata dari kateglo dan sinonimkata.com.
2. Menggunakan lingkungan ujicoba yang lebih baik lagi dalam segi hardware computer dan menggunakan multi threading saat pembentukan sinonim kata menggunakan kamus kata kateglo, karena proses eksekusi yang lama dalam penghitungan cosine similarity antar kata.

DAFTAR PUSTAKA

- [3] Wikipedia. Wikipedia. [Online].
<https://id.wikipedia.org/wiki/Tokenisasi>
- [5] Hikaryuki. Hikaryuki. [Online].
http://static.hikaruyuuki.com/wp-content/uploads/stopword_list_tala.txt
- [2] Kementrian Riset dan Teknologi Republik Indonesia. ristek.go.id. [Online].
<http://www1.ristek.go.id/?module=News%20News&id=8705>
- [4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, *Introduction to Information Retrieval*.: Cambridge University Press, 2008.
- [1] DIKTI. dikti.go.id. [Online].
<http://www.dikti.go.id/kolokium-di-australia-kerjasama-antar-penelitisemakin-dibutuhkan-di-indonesia/>
- [6] Olivier Michel, Pierre Comon and Alfred O. Hero Laurent Galluccio, "Graph based k-means clustering," *Graph based k-means clustering*, p. 15, 2011.
- [7] Wikipedia. Prim Algorithms Wikipedia. [Online].
https://en.wikipedia.org/wiki/Prim's_algorithm
- [8] Wikipedia. Dunn Index - Wikipedia. [Online].
https://en.wikipedia.org/wiki/Dunn_index
- [9] Wikipedia. Betweenness Centrality - Wikipedia. [Online].
https://en.wikipedia.org/wiki/Betweenness_centrality#Model_networks
- [10] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,," 2003.
- [11] Annisa S. Febriana. (2010, January) Ivan Lanin: Indonesian Language Evangelist. [Online].

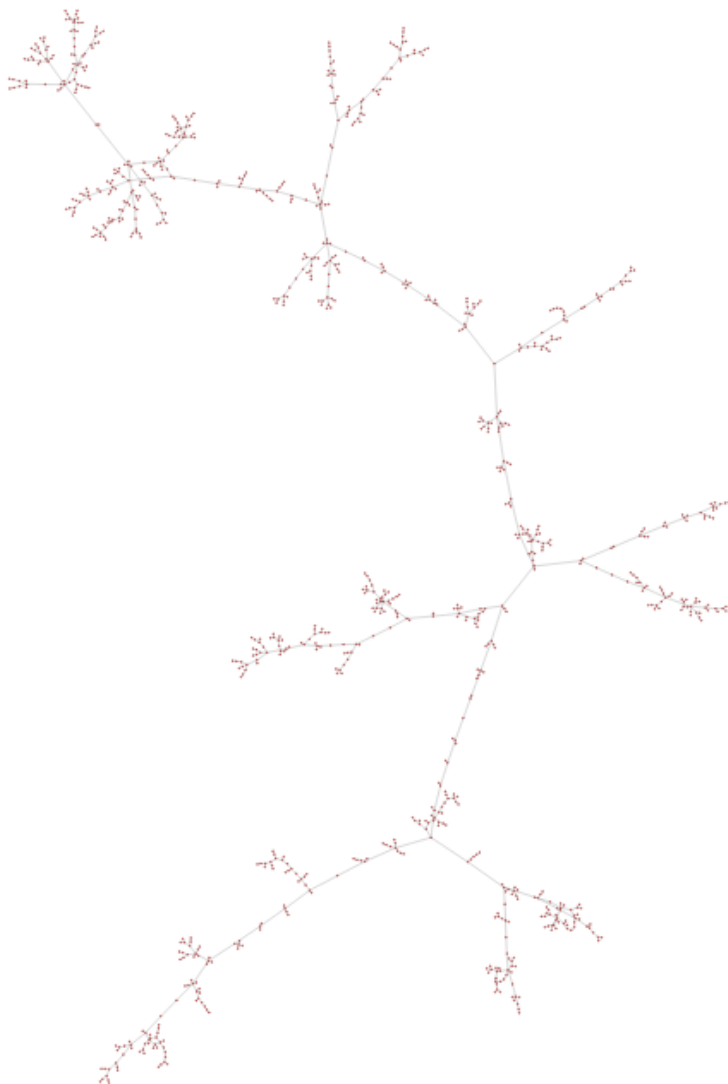
www.thejakartapost.com/news/2010/01/25/ivan-lanin-indonesian-language-evangelist.html

LAMPIRAN

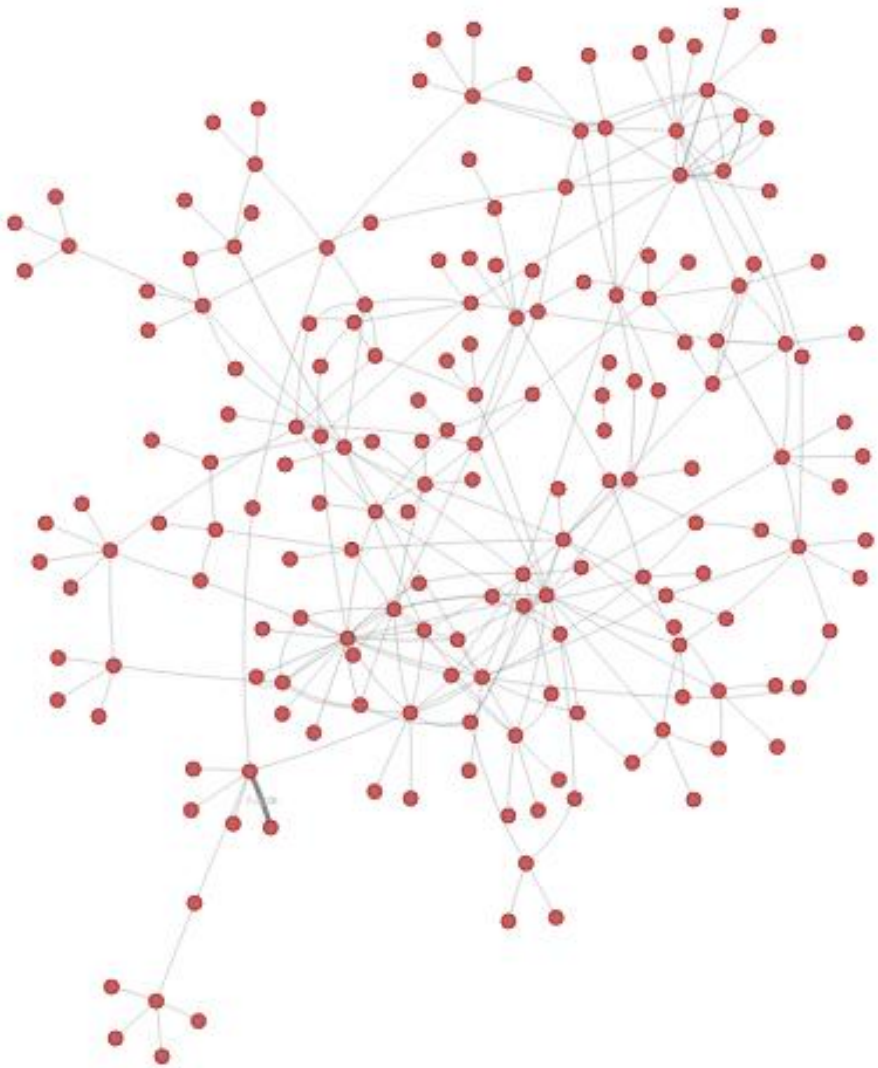
8.1 Lampiran Gambar



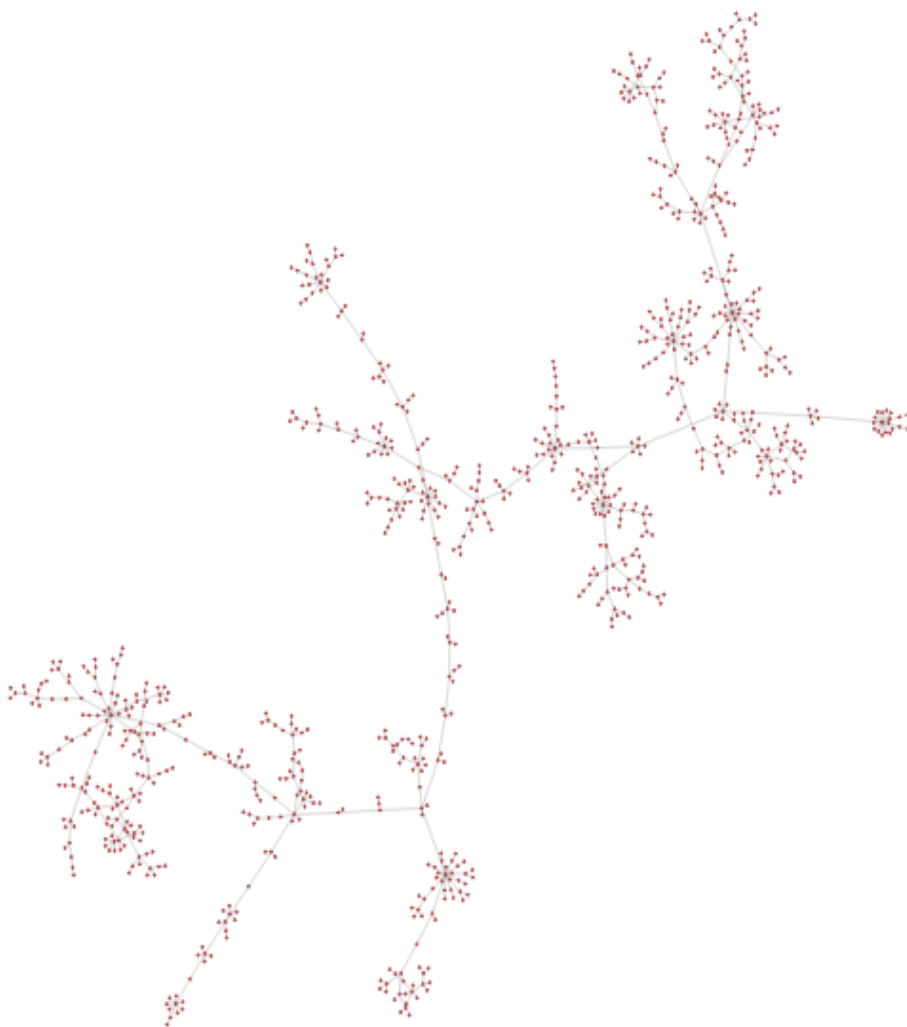
Gambar 8.1 Gambar graf kerjasama peneliti FTIF ITS sebelum diterapkan metode clustering



Gambar 8.2 Gambar minimum spanning tree graf kerjasama peneliti ITS

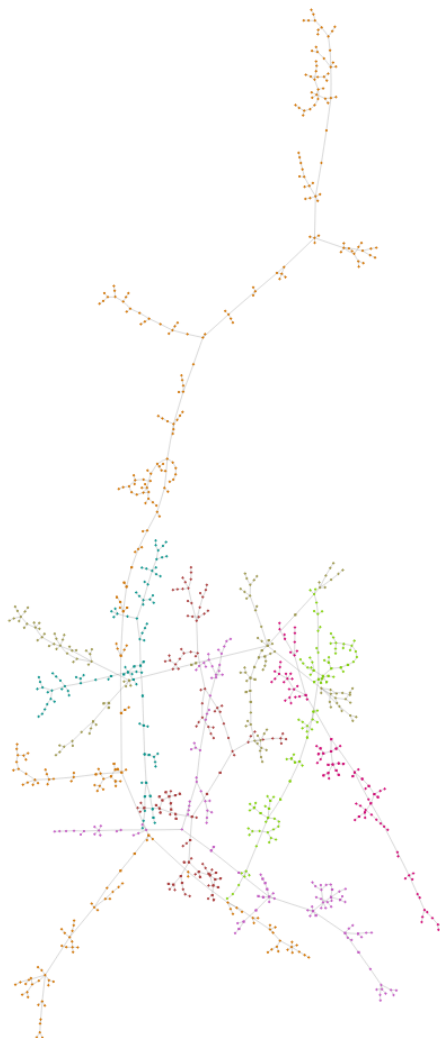


Gambar 8.3 Gambar graf kerjasama peneliti FTIF ITS ekspansi sinonim kata

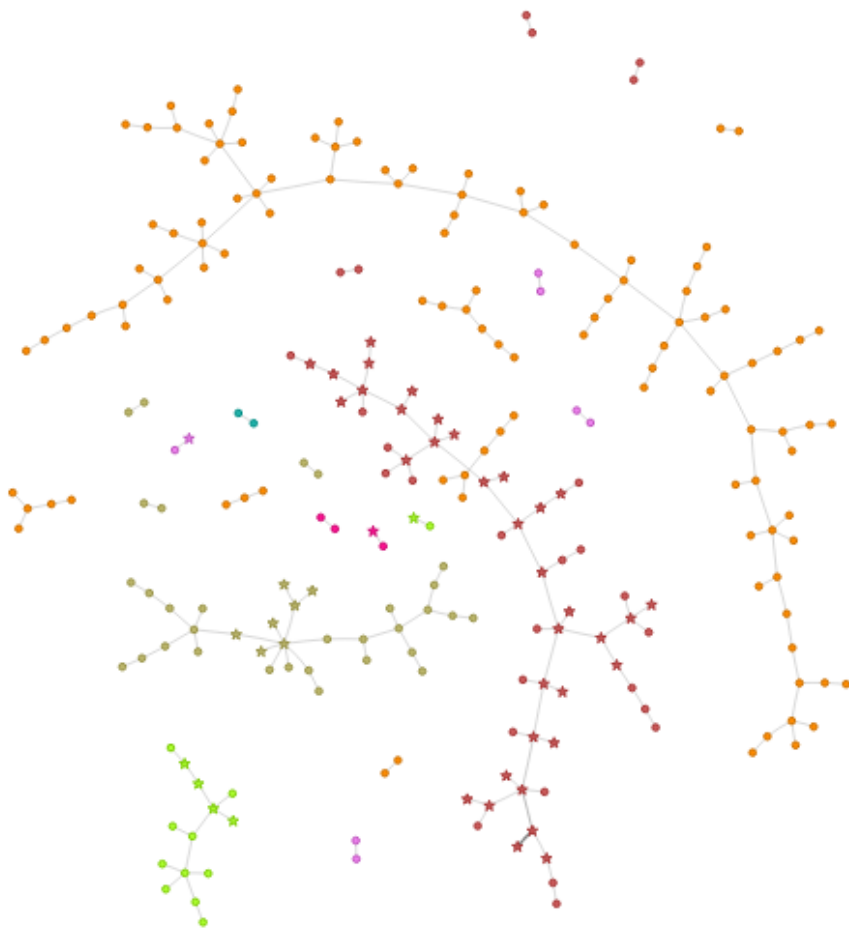


Gambar 8.4 Gambar minimum spanning tree yang terbentuk dari graf kerjasama peneliti ITS ekspansi sinonim kata

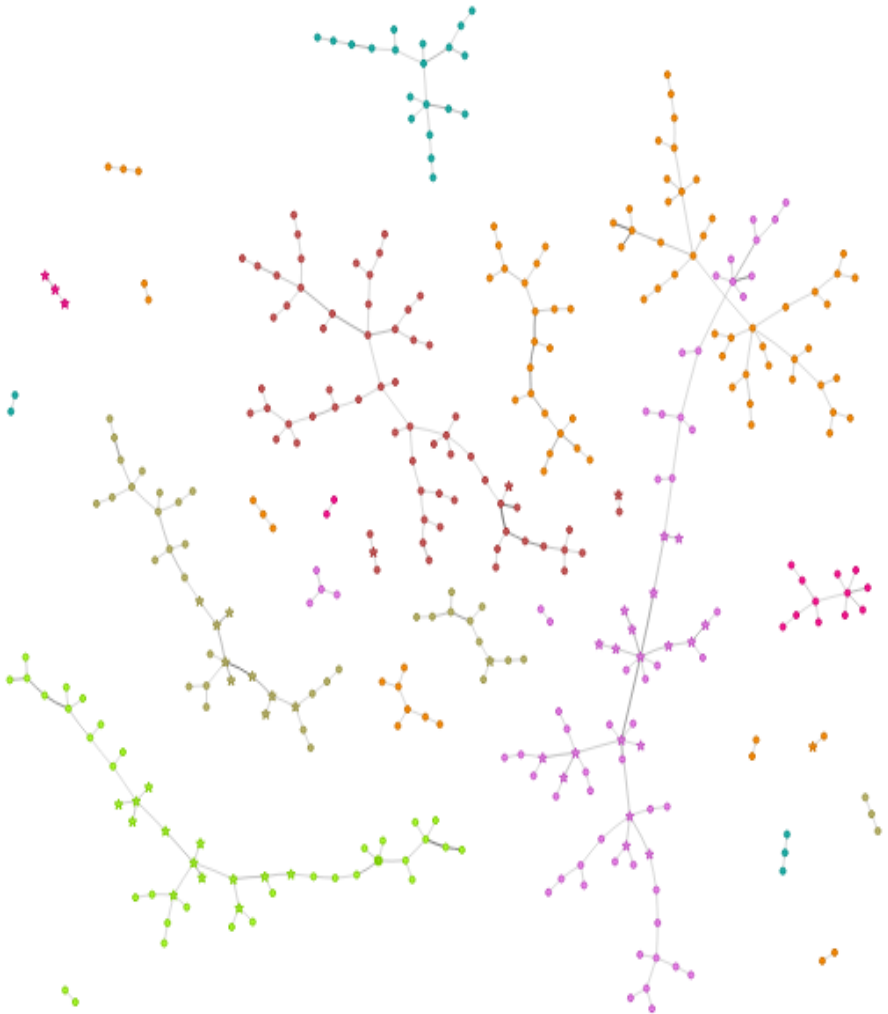
8.1.1 Gambar Graf Cluster Kerjasama Peneliti dengan Nilai K sama dengan 7



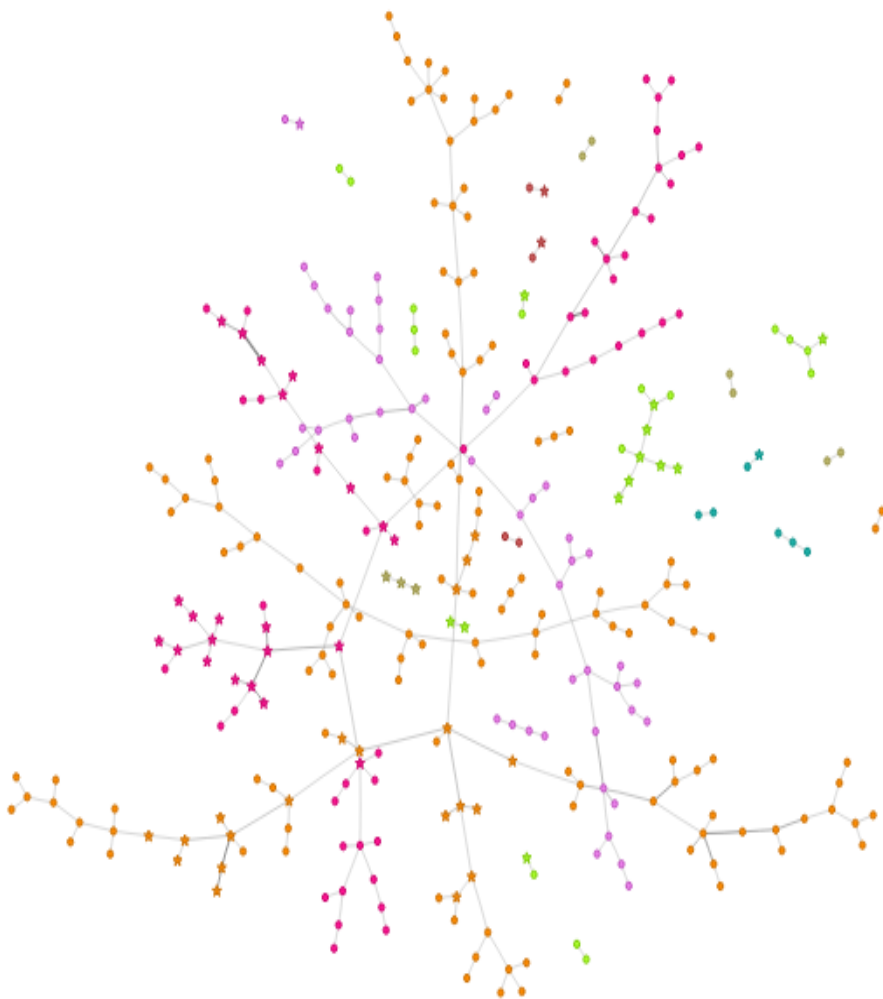
Gambar 8.5 Gambar graf cluster peneliti ITS dengan K sama dengan 7



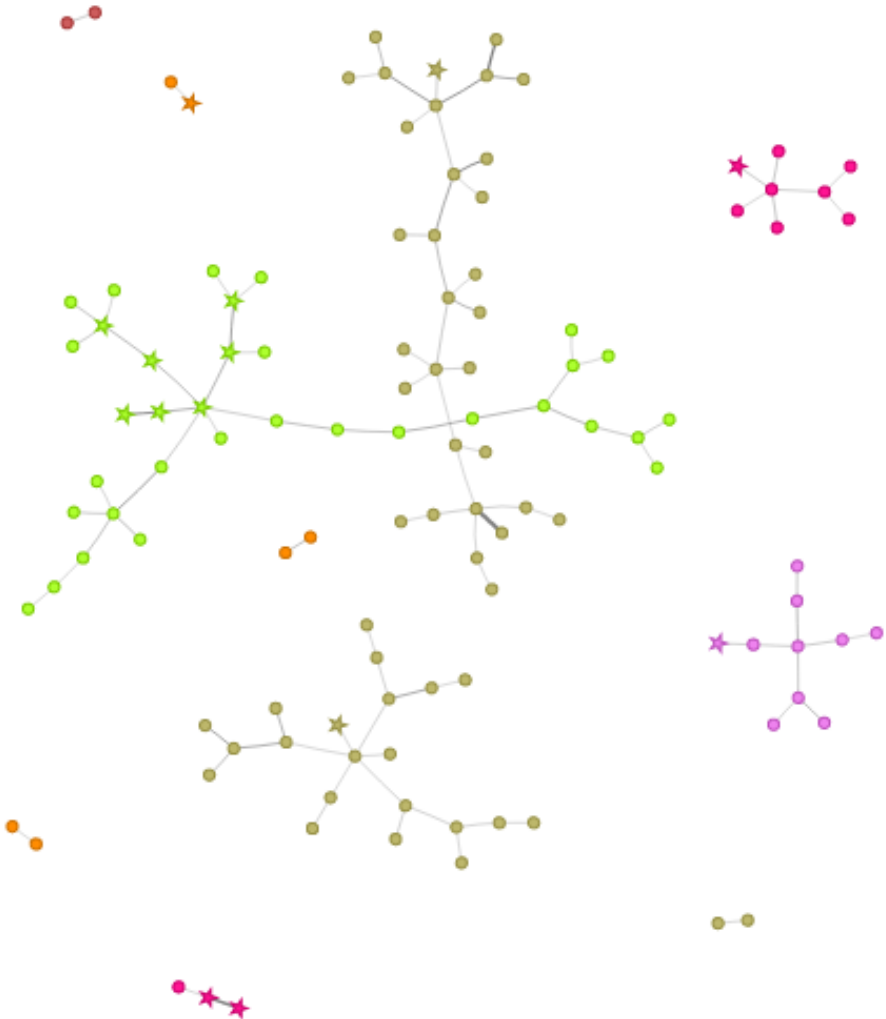
Gambar 8.6 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 7



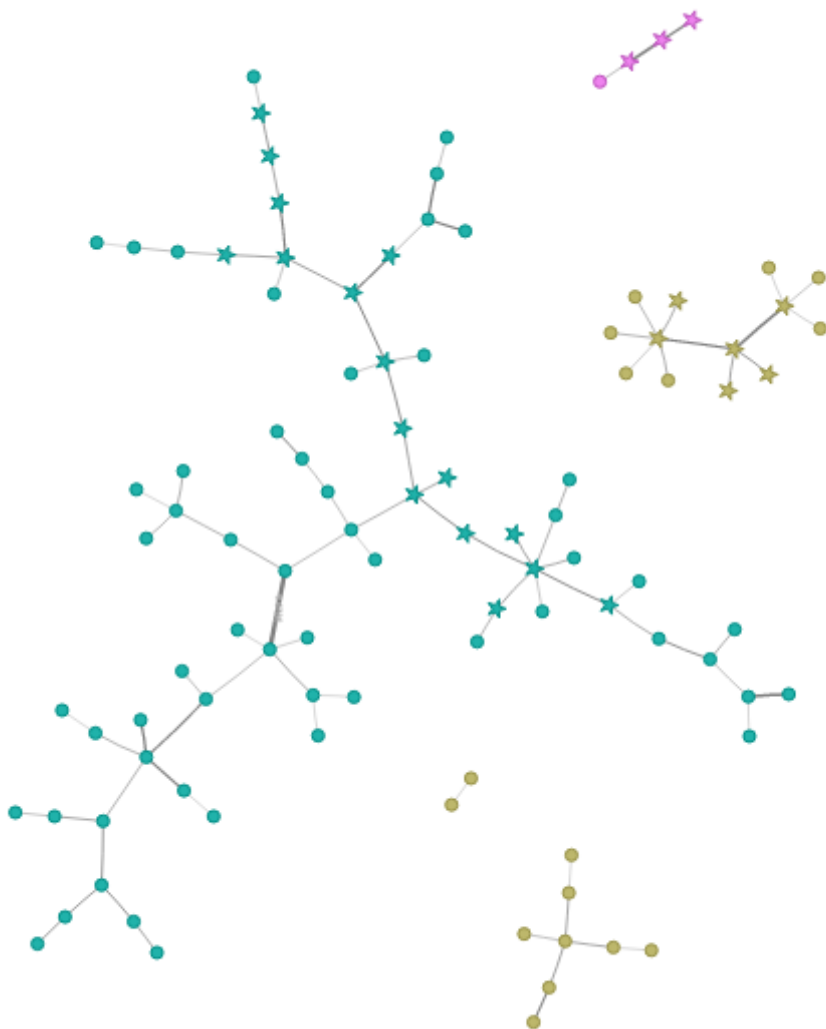
Gambar 8.7 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 7



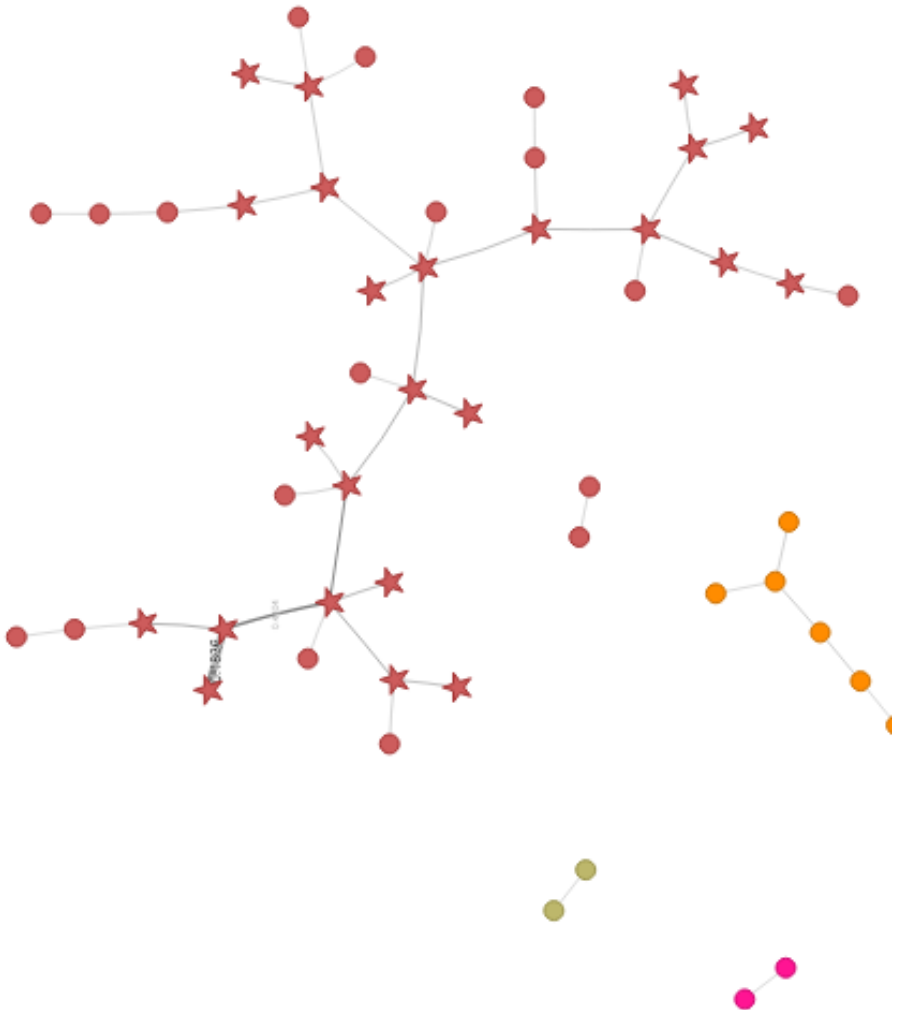
Gambar 8.8 Gambar graf peneliti FTSP ITS dengan K sama dengan 7



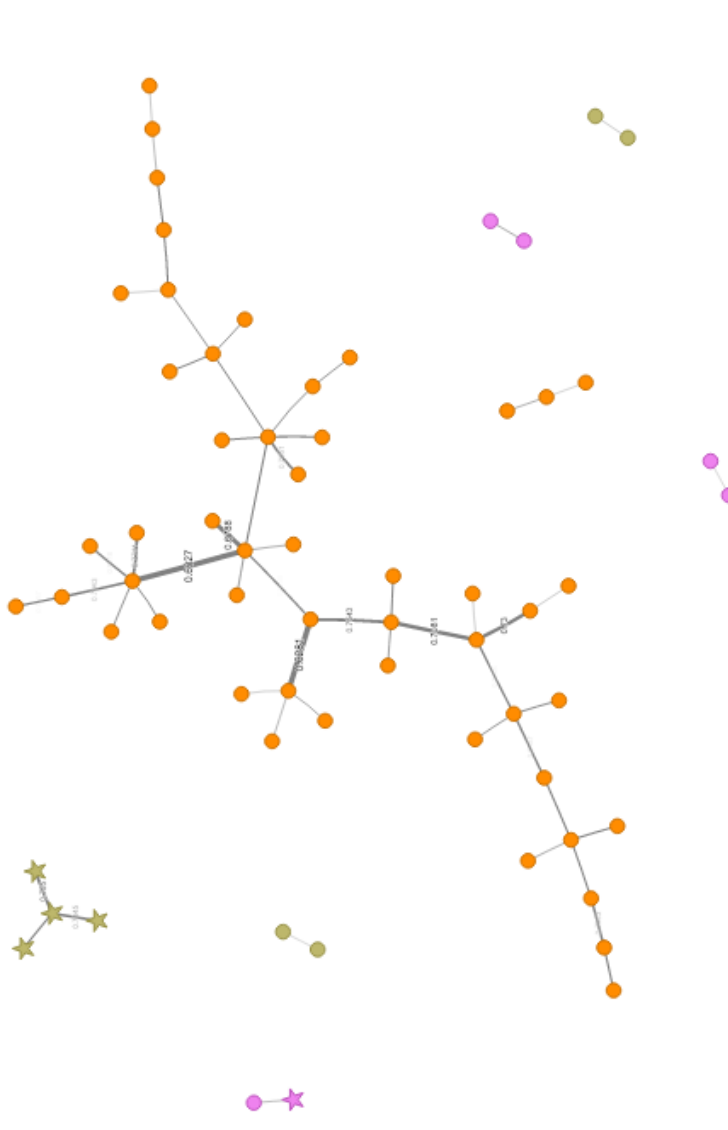
Gambar 8.9 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 7



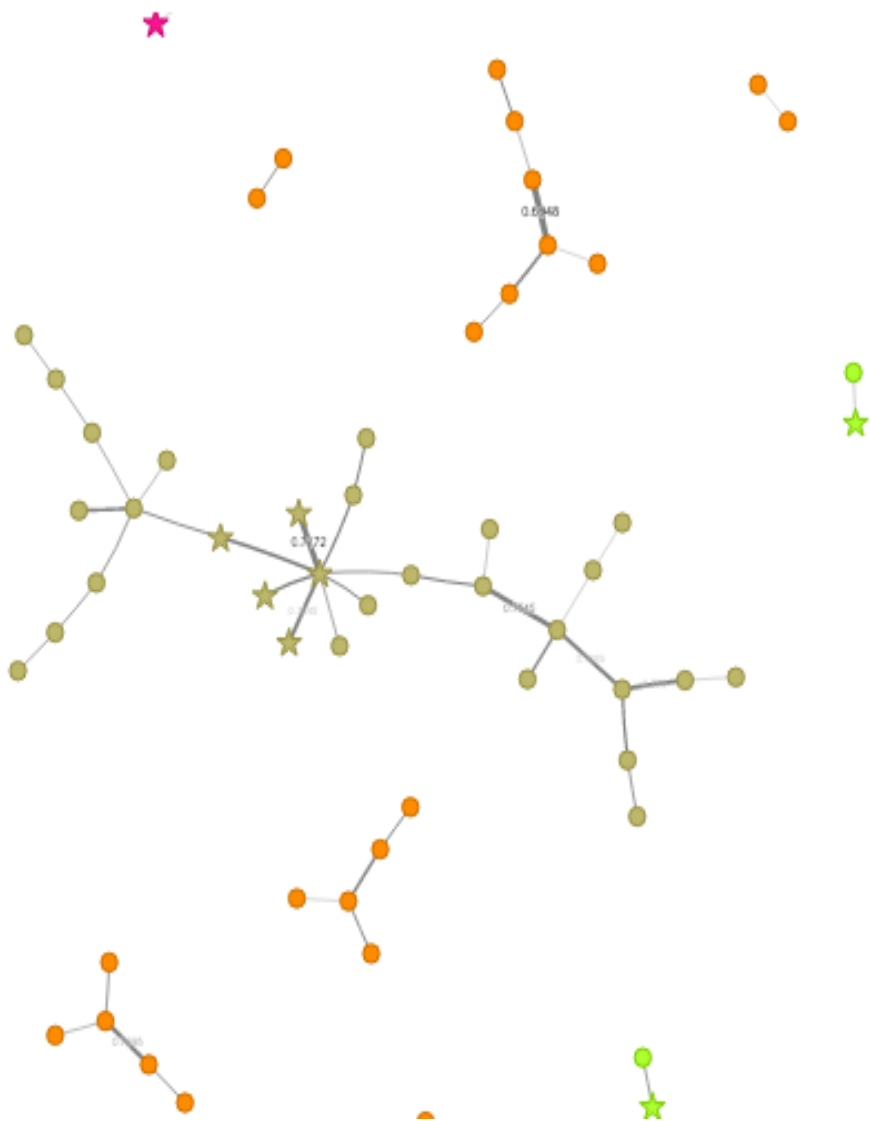
Gambar 8.10 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 7



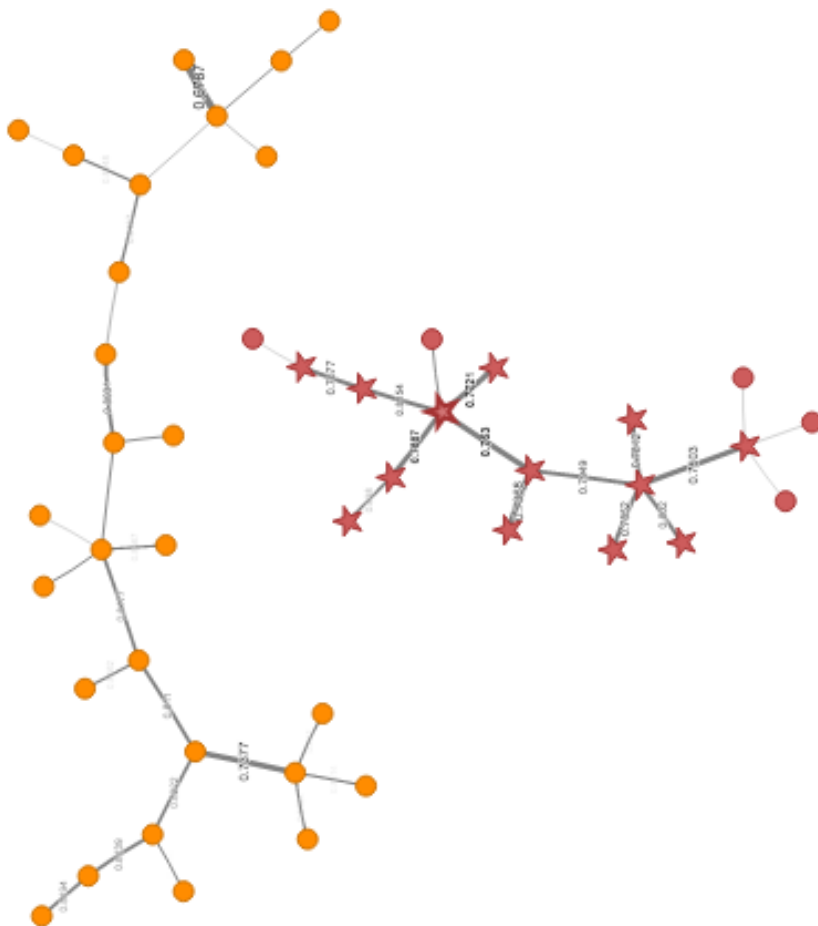
Gambar 8.11 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 7



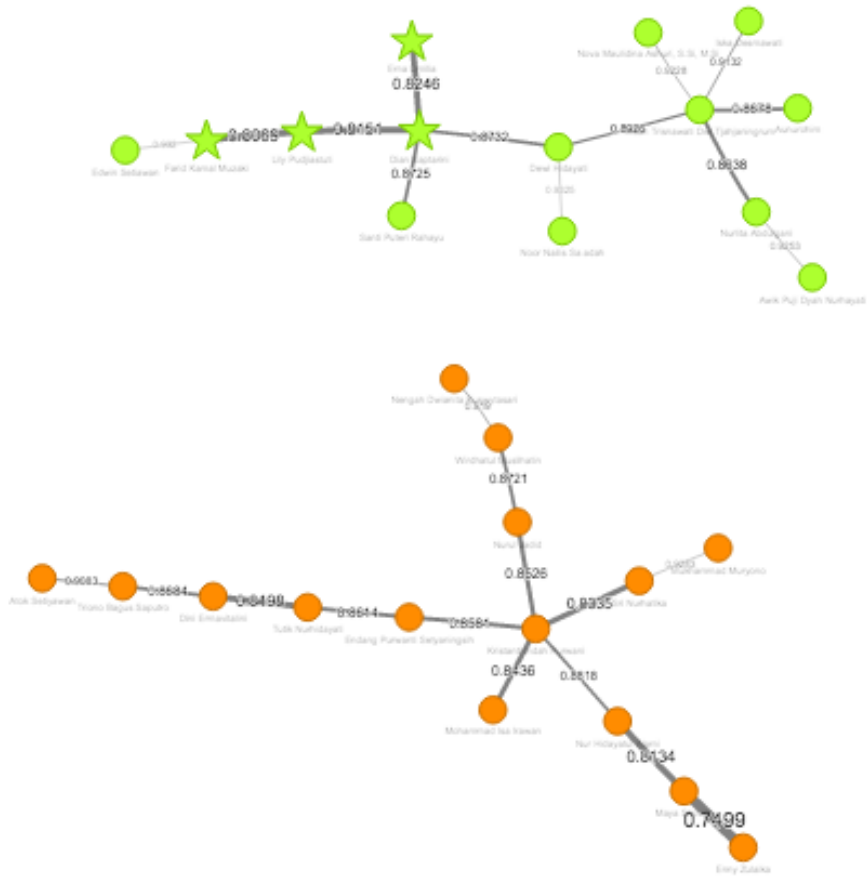
Gambar 8.12 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 7



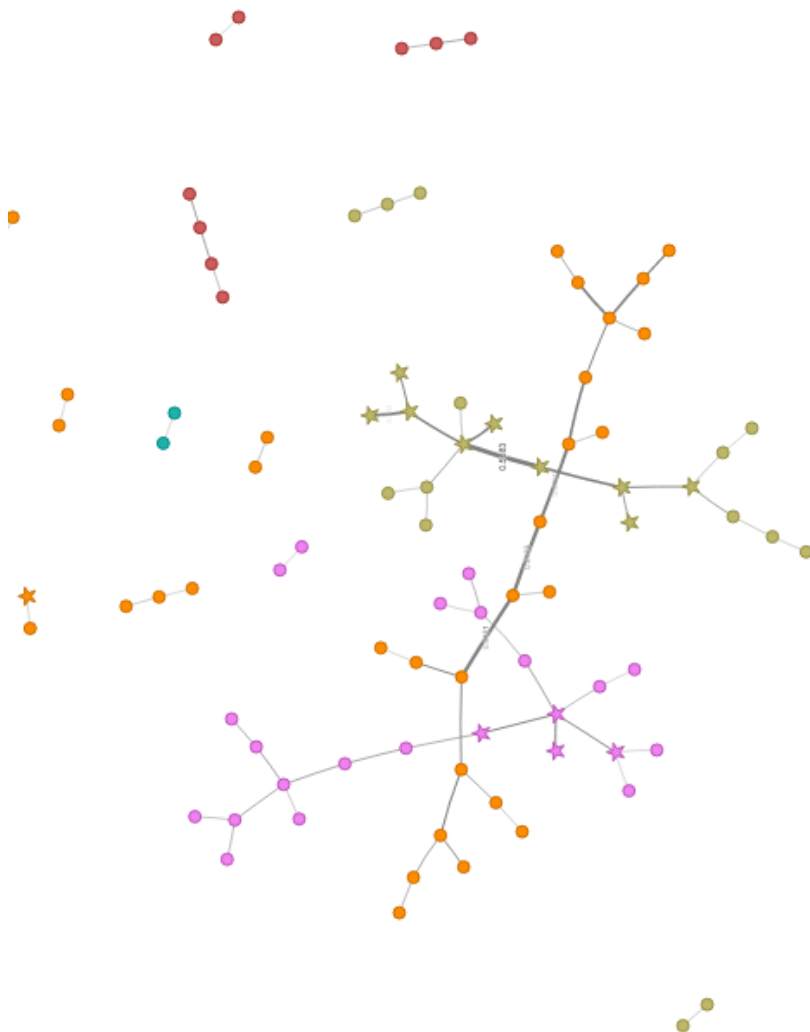
Gambar 8.13Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 7



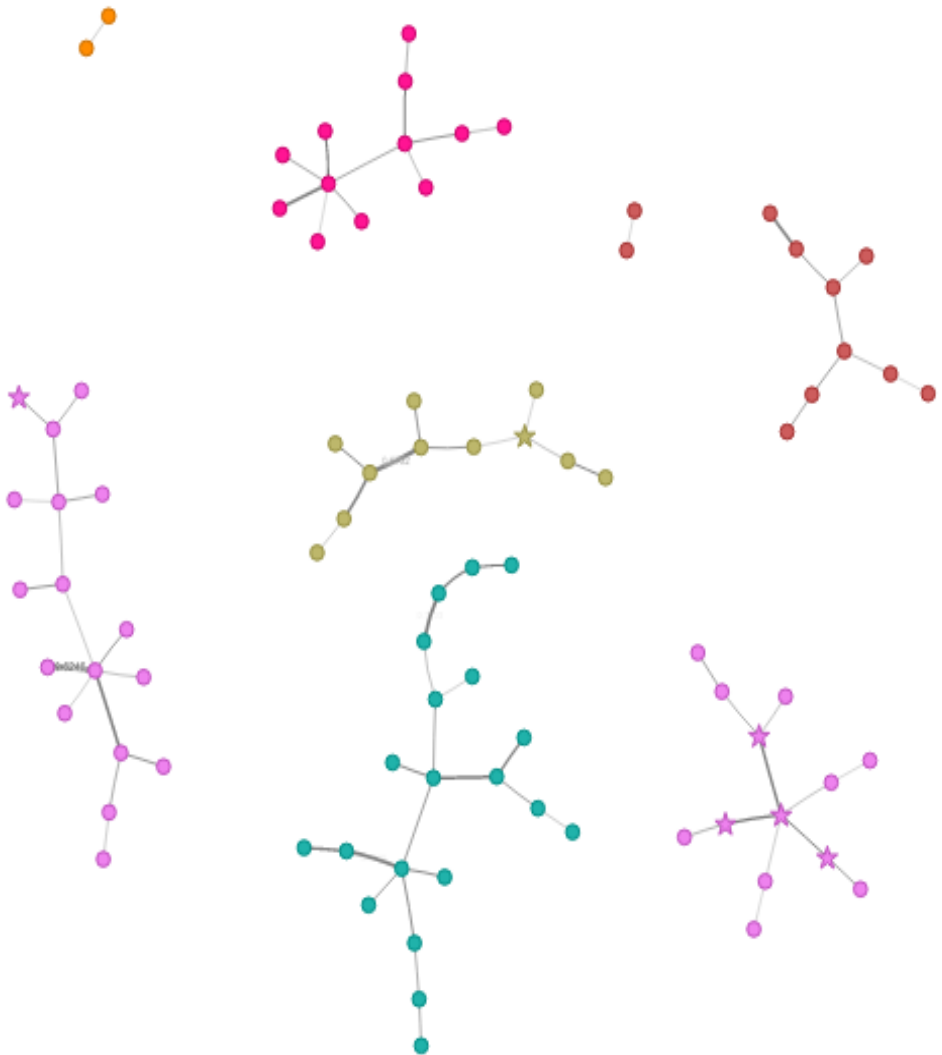
Gambar 8.14 Gambar graf cluster peneliti jurusan Kimia ITS dengan nilai K sama dengan 7



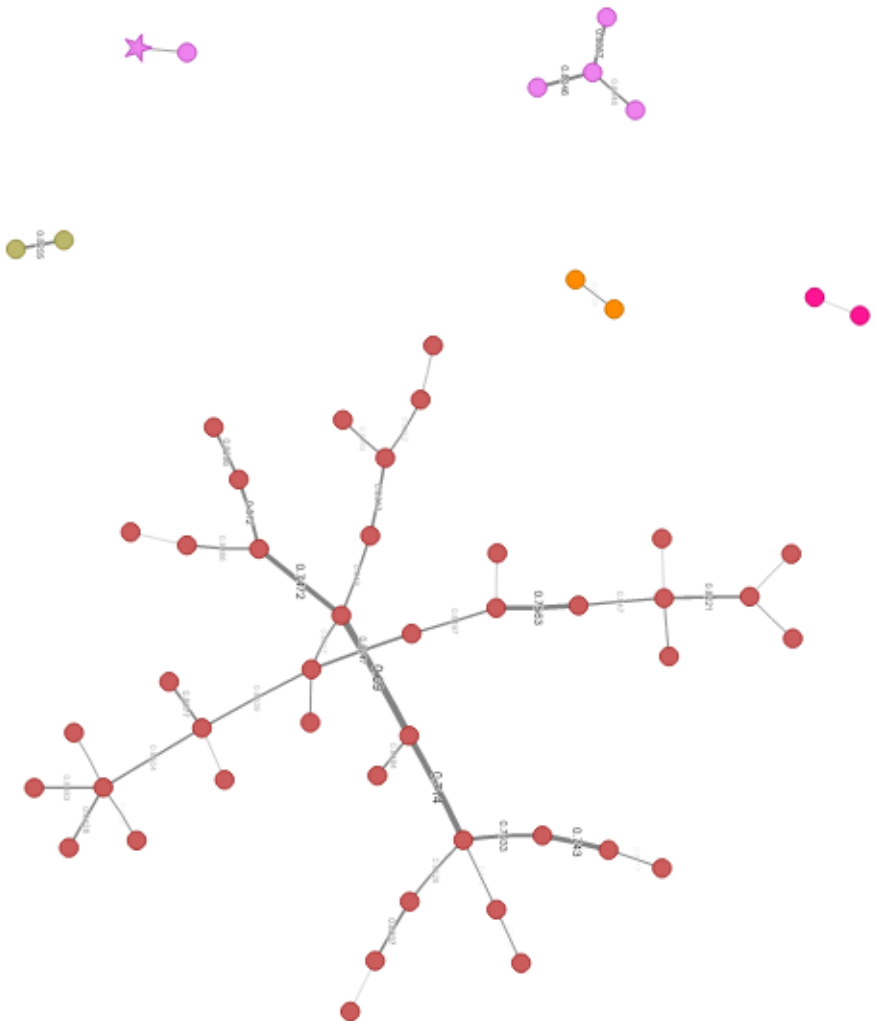
Gambar 8.15 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 7



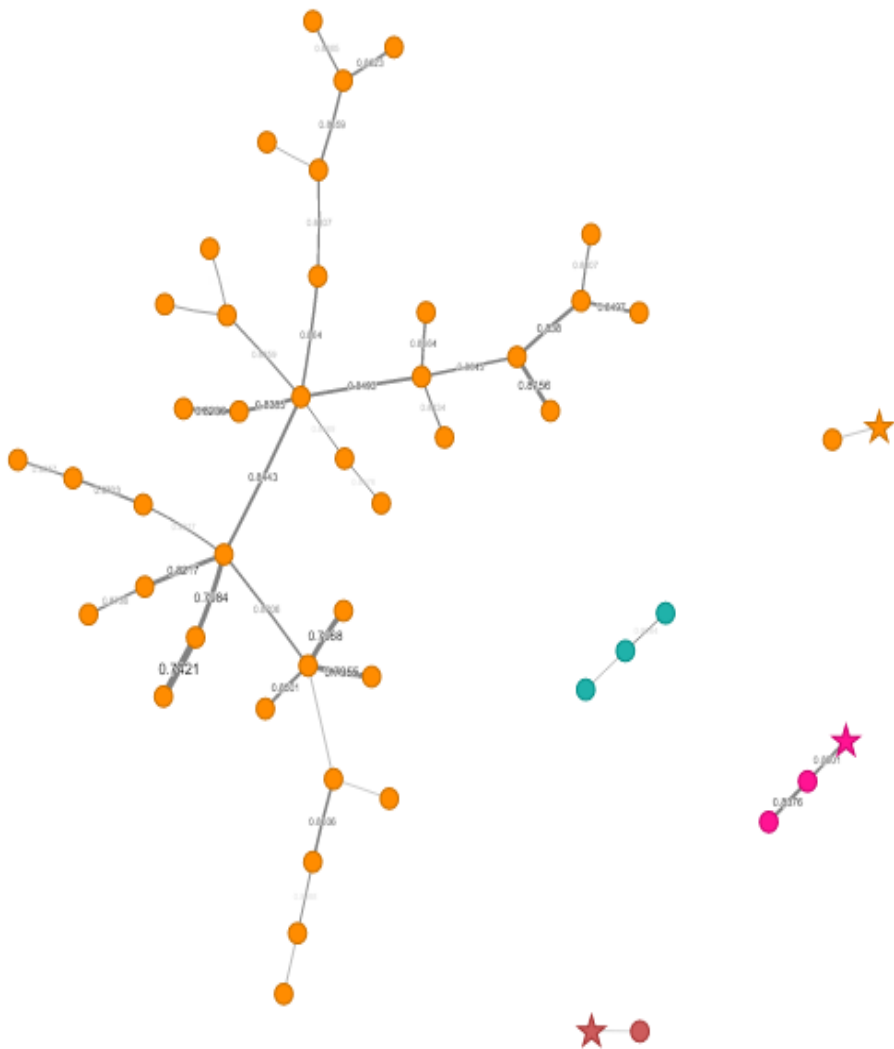
Gambar 8.16 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 7



Gambar 8.17 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 7



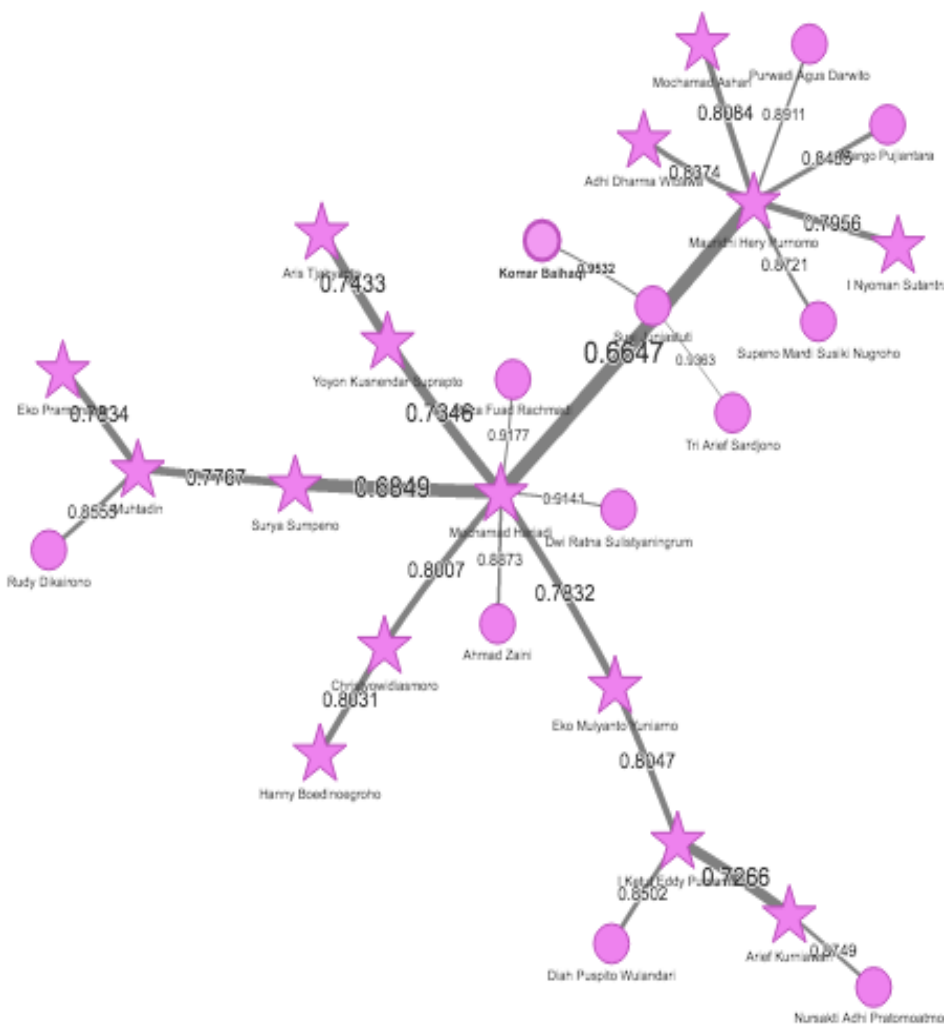
Gambar 8.19 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 7



Gambar 8.20 Gambar graf cluster peneliti jurusan Teknik Industri ITS dengan nilai K sama dengan 7



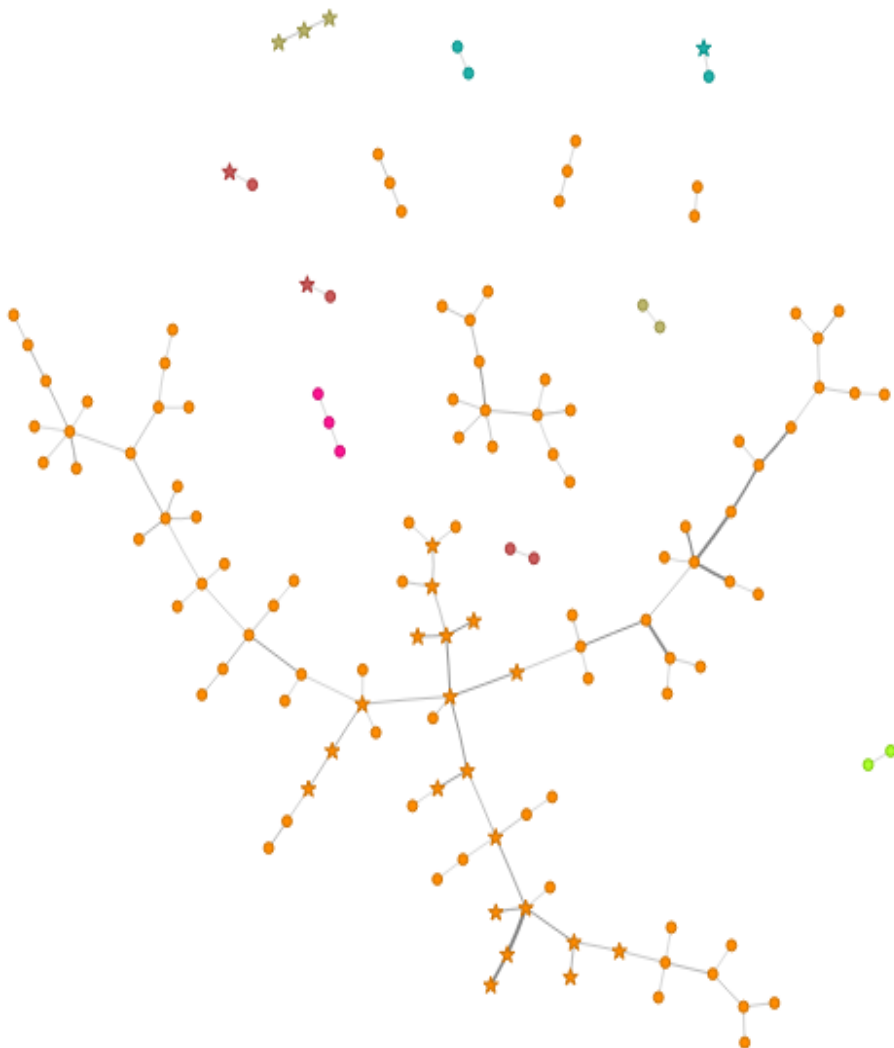
Gambar 8.21 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 7



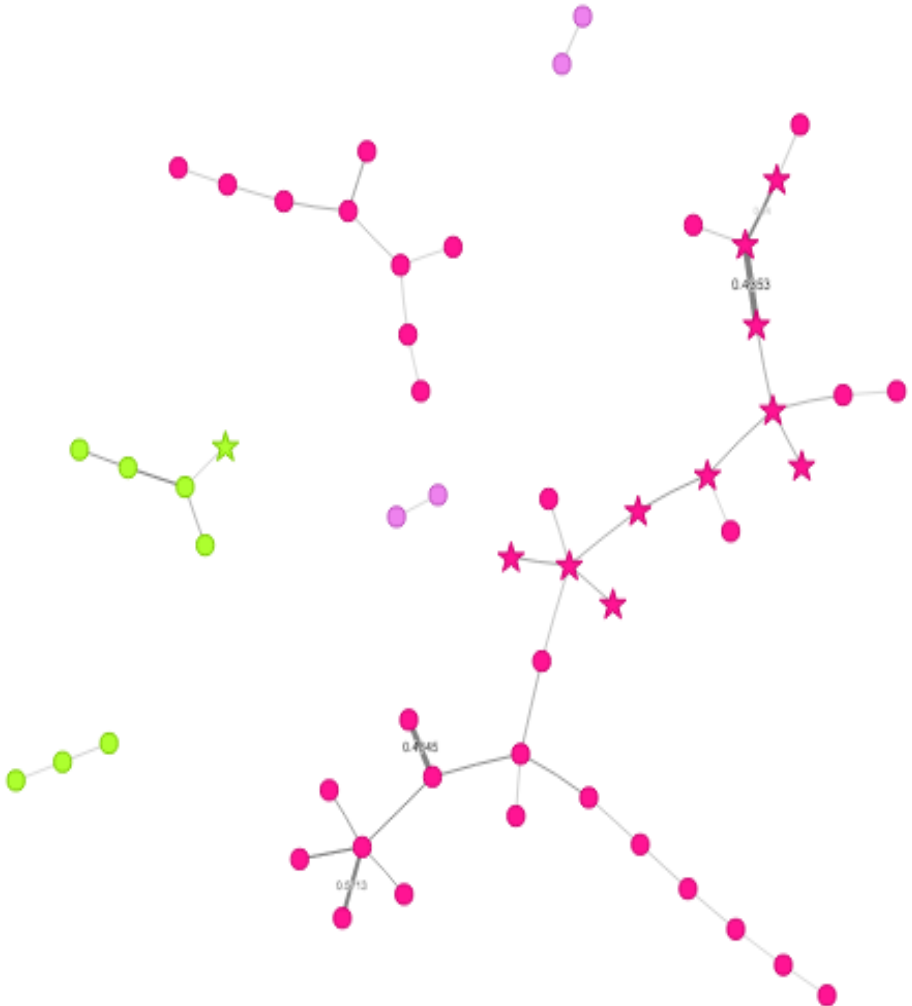
Gambar 8.22 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 7



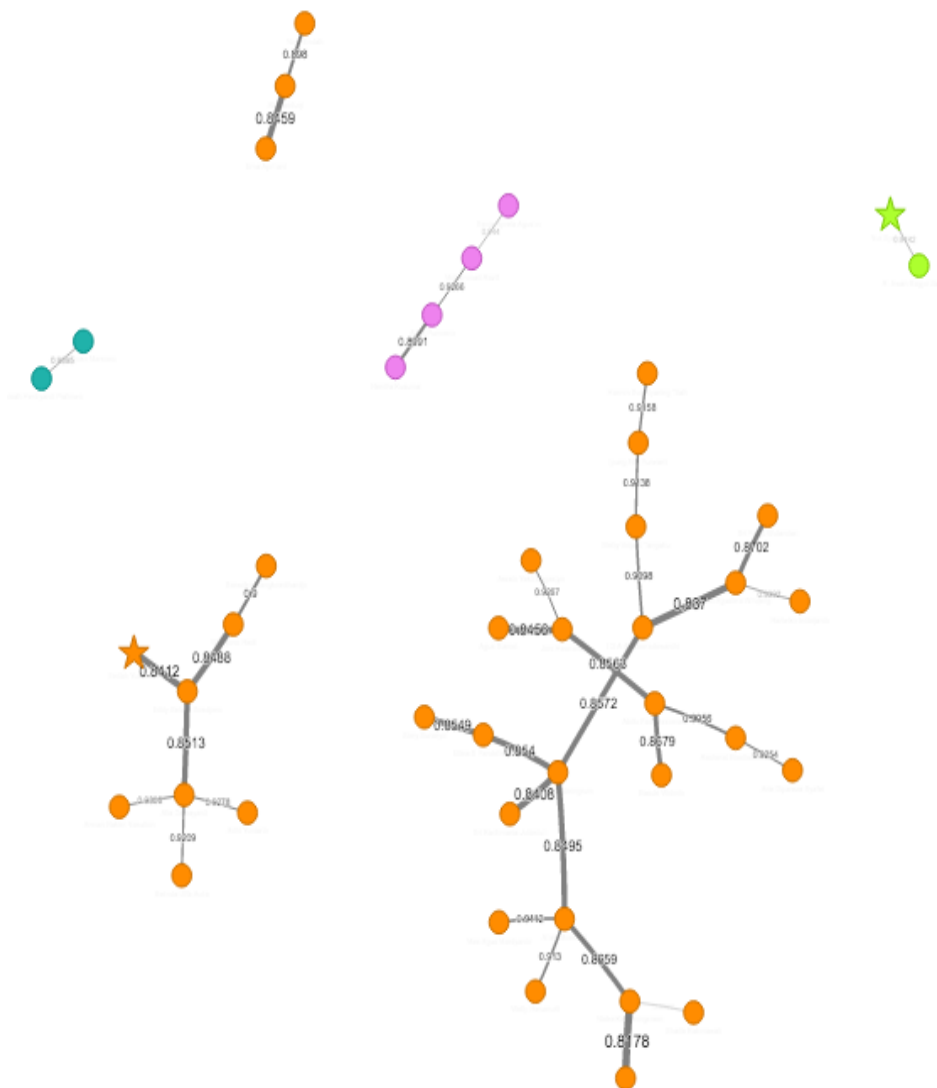
Gambar 8.23 Gambar graf cluster peneliti jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 7



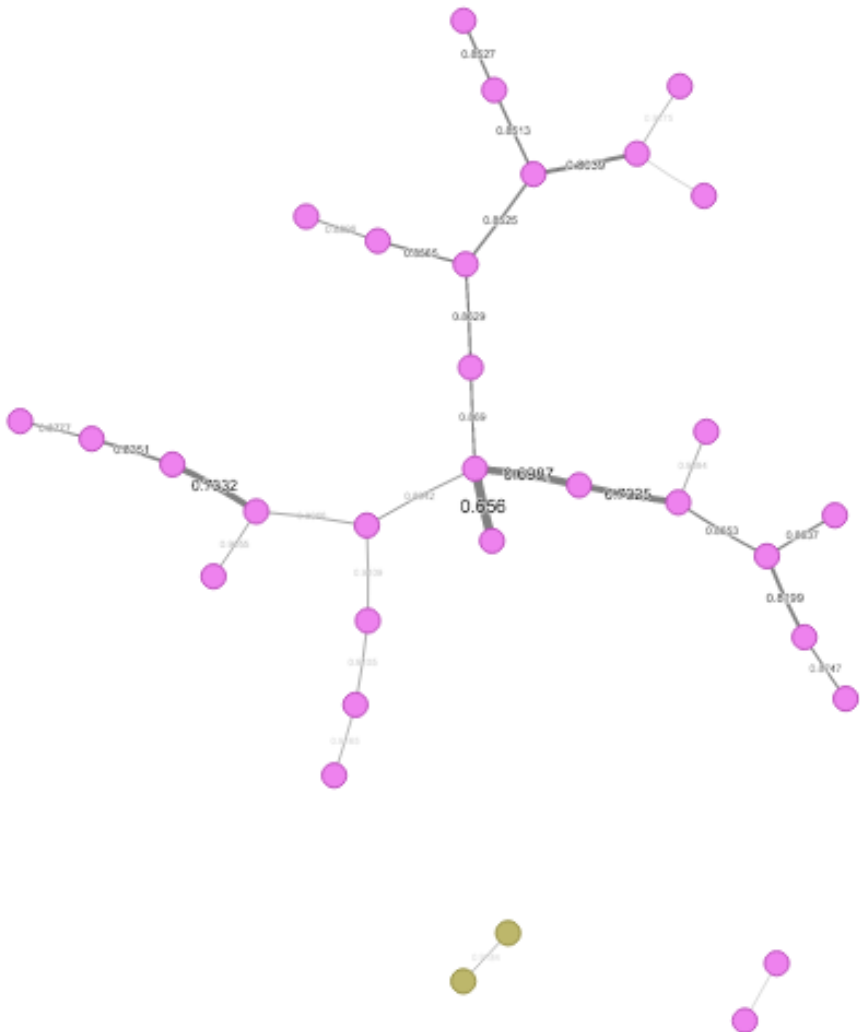
Gambar 8.24 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 7



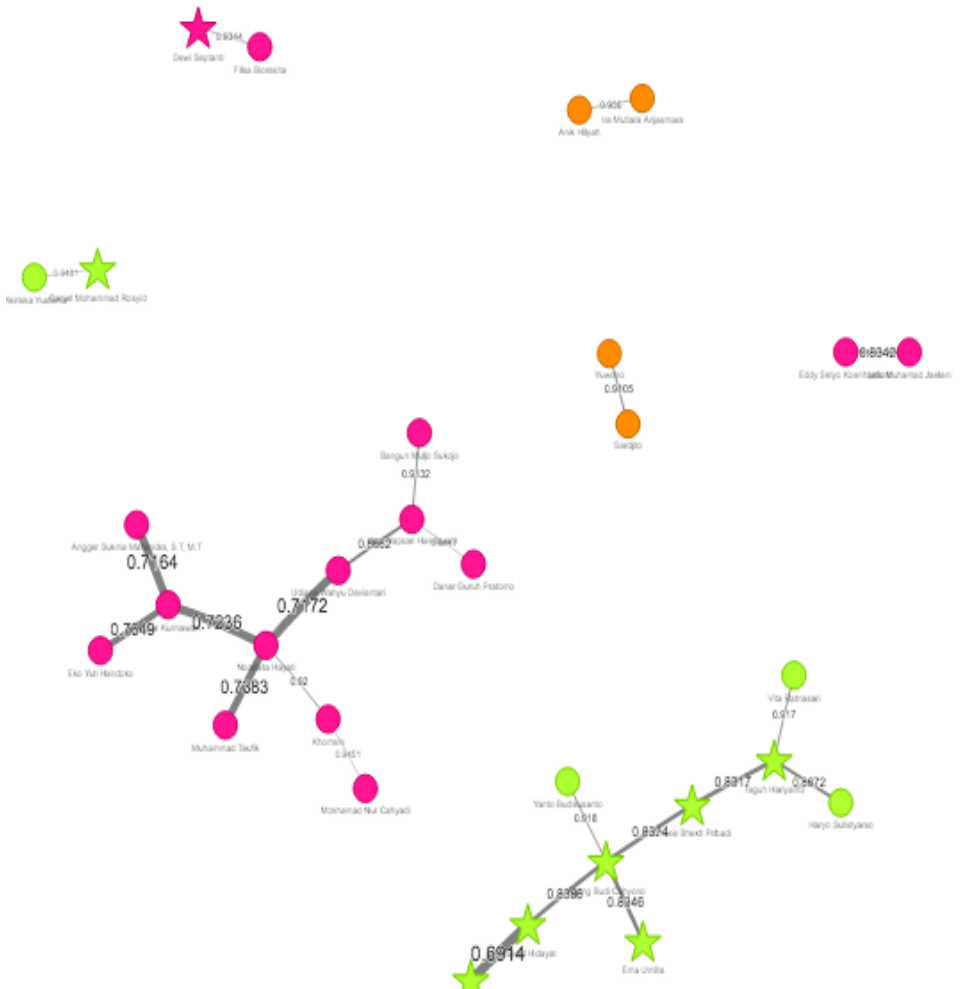
Gambar 8.25 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 7



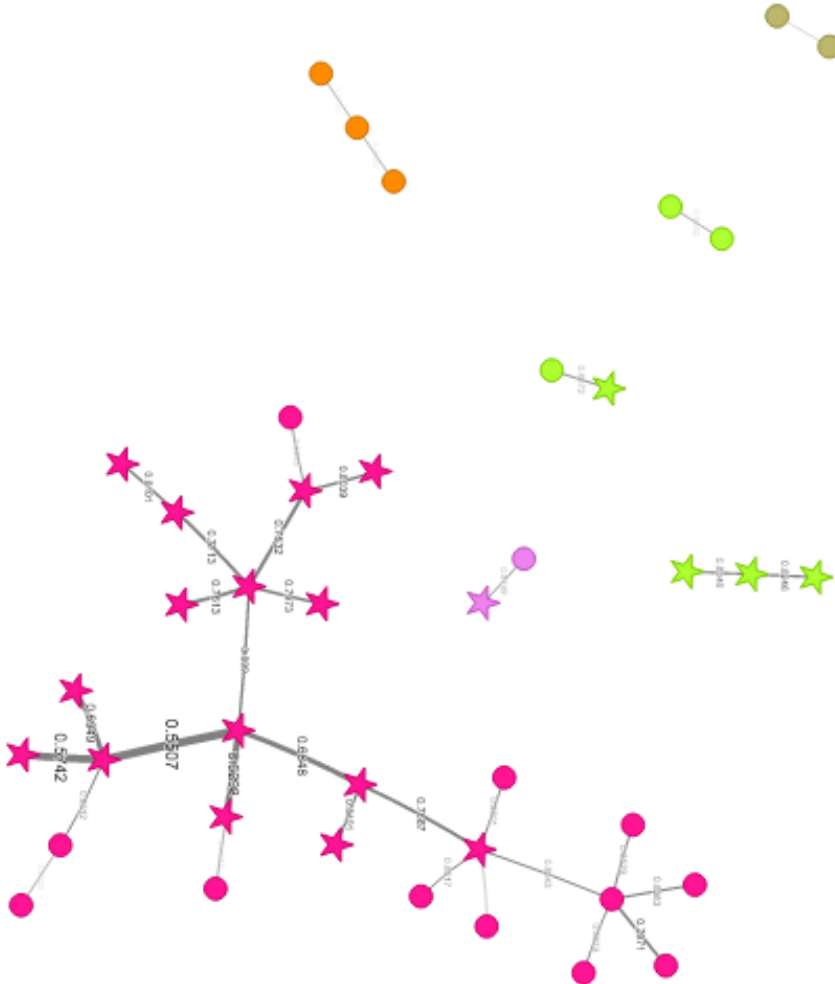
Gambar 8.26 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 7



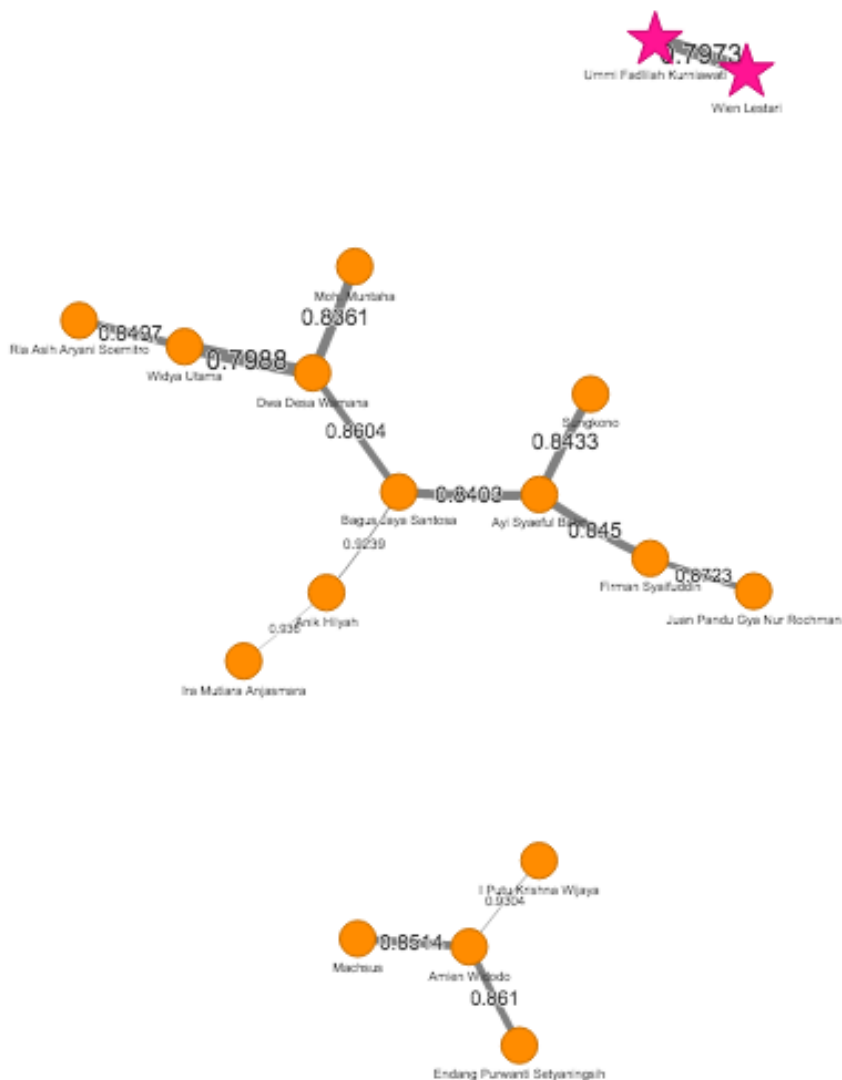
Gambar 8.27 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 7



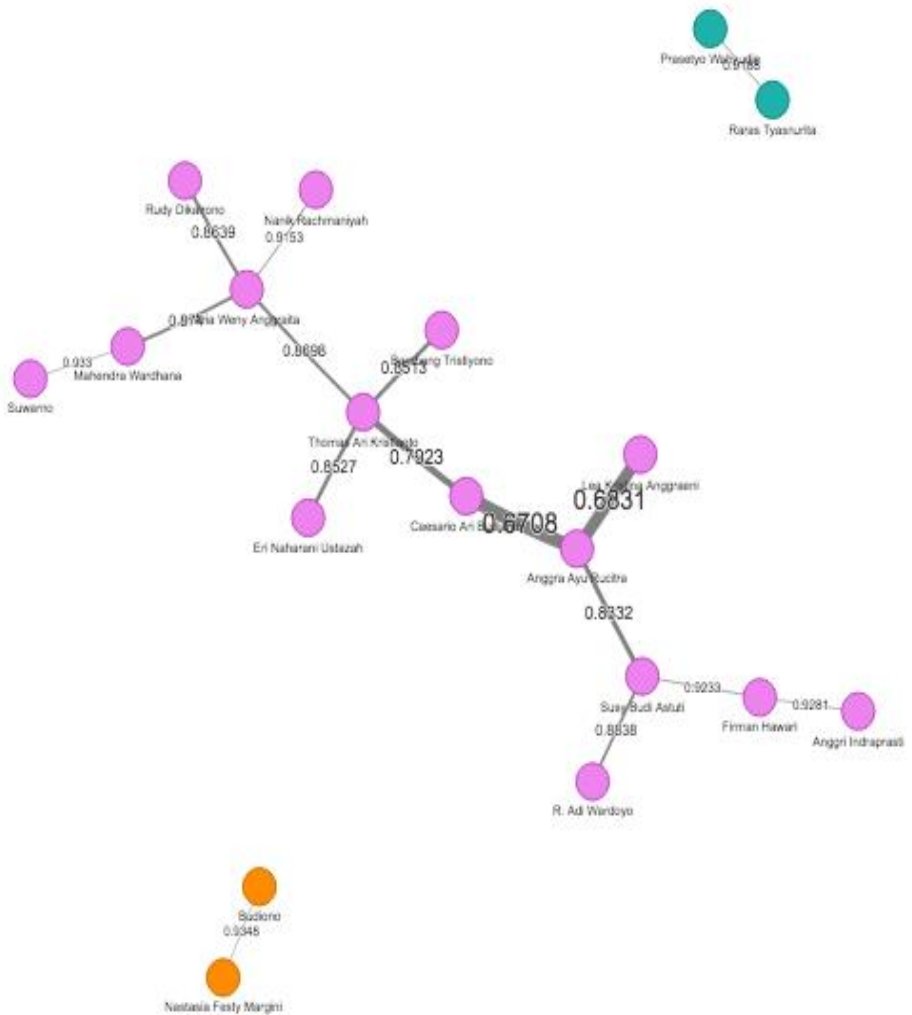
Gambar 8.28 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 7



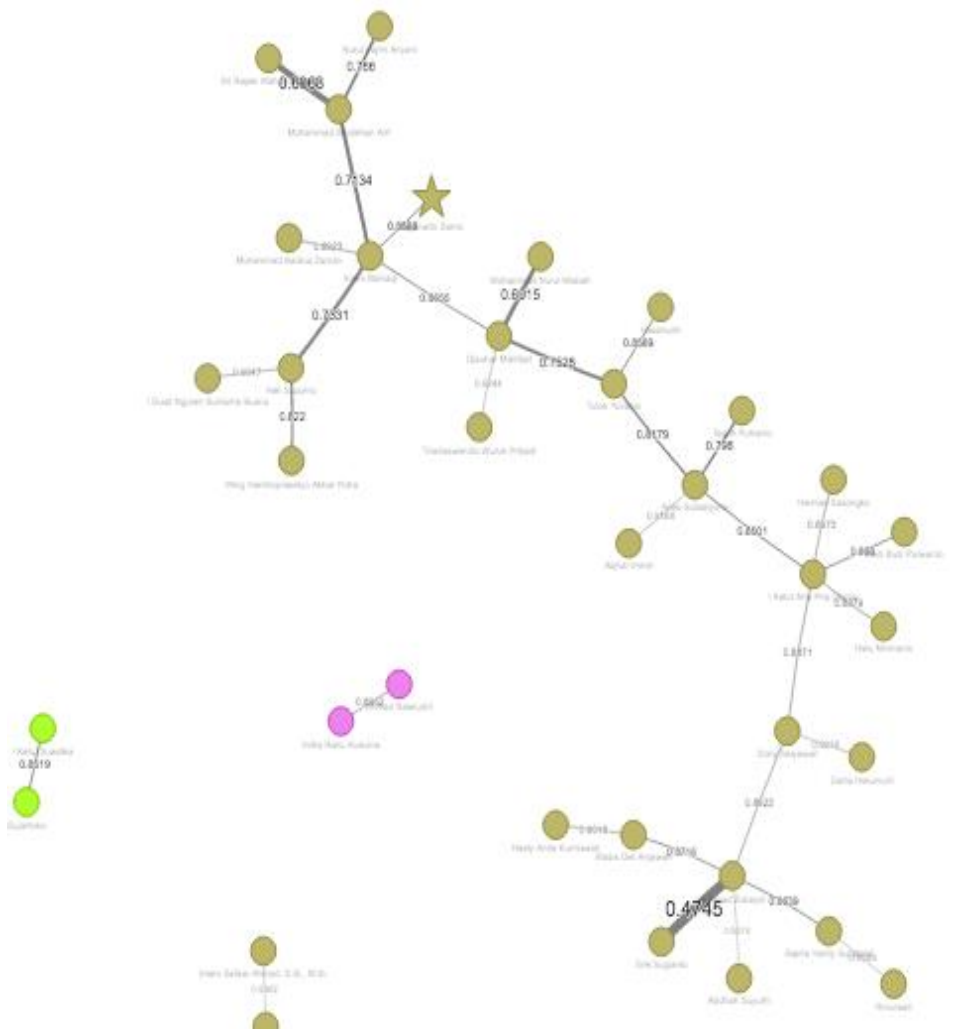
Gambar 8.29 Gambar graf cluster peneliti jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 7



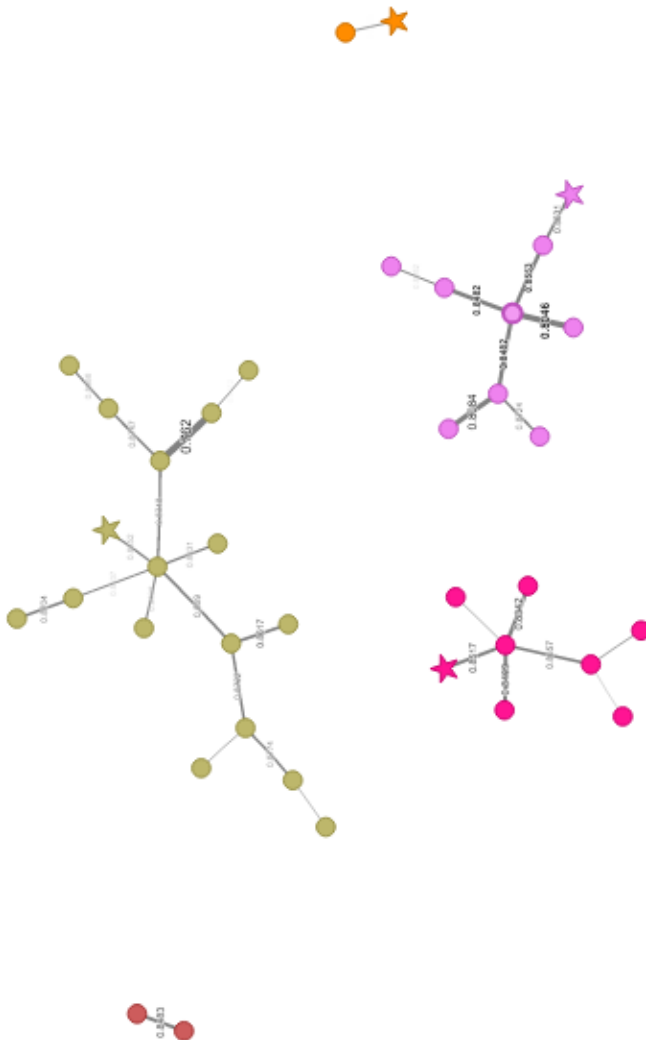
Gambar 8.30 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 7



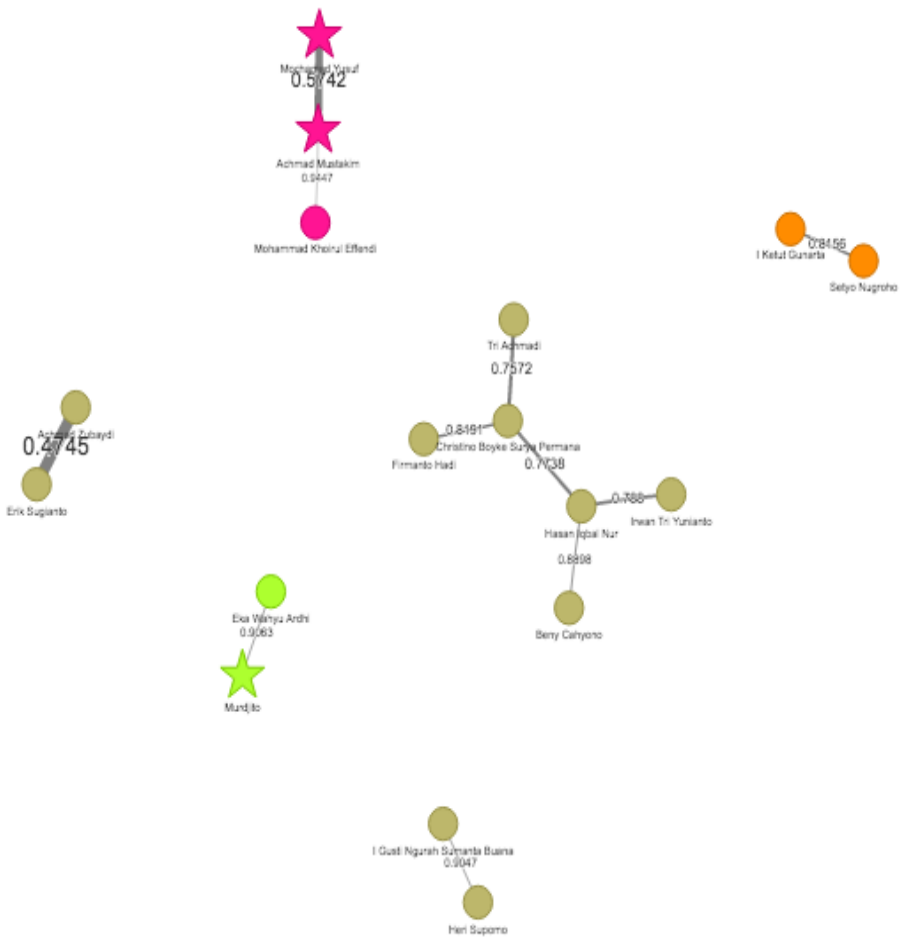
Gambar 8.31 Gambar graf cluster peneliti jurusan Desain Interior ITS dengan nilai K sama dengan 7



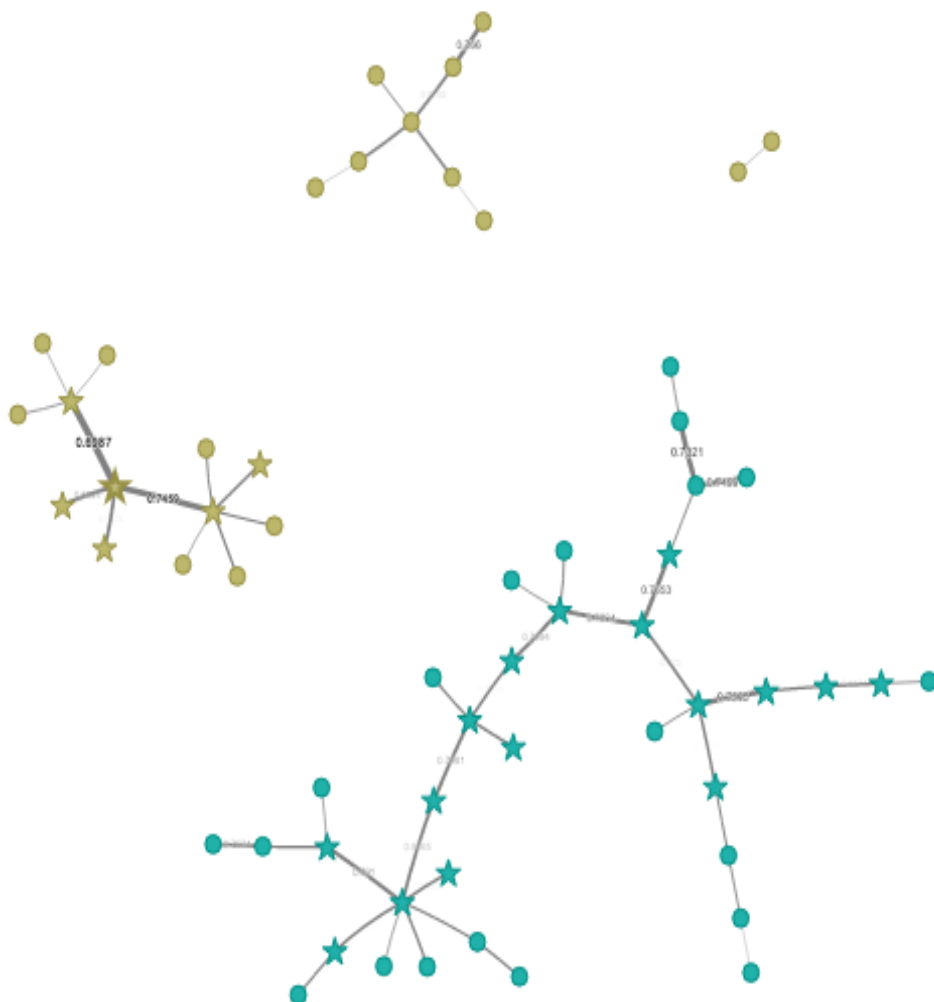
Gambar 8.32 Gambar graf cluster peneliti jurusan Teknik Perkapalan ITS dengan nilai K sama dengan 7



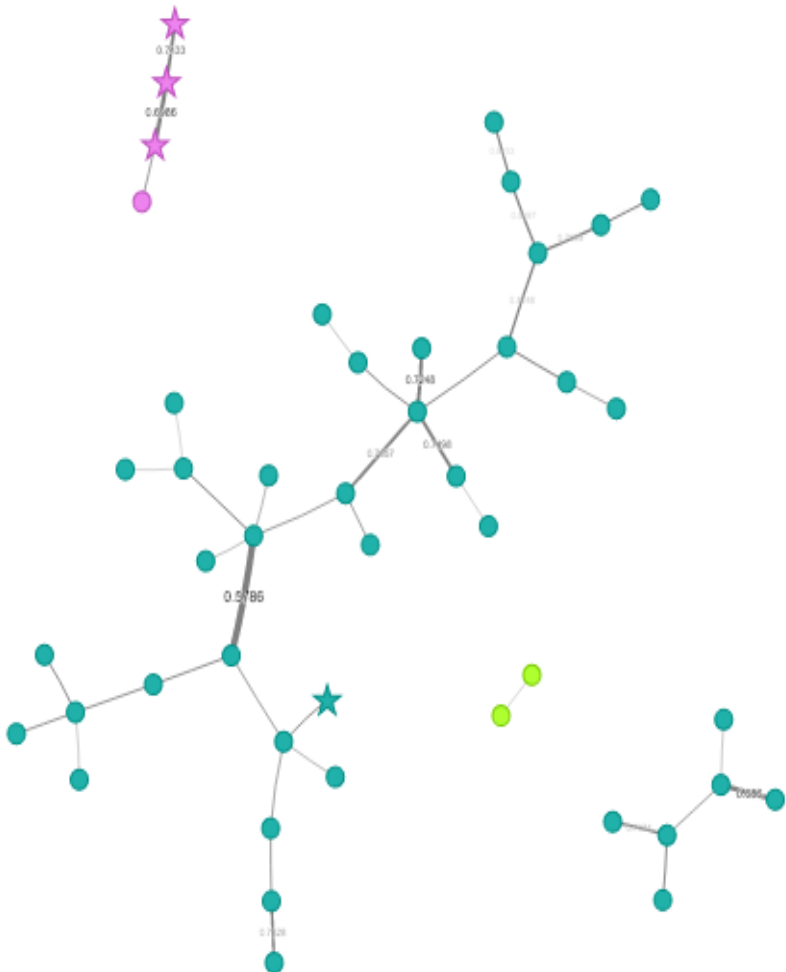
Gambar 8.33 Gambar graf cluster peneliti jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 7



Gambar 8.35 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 7

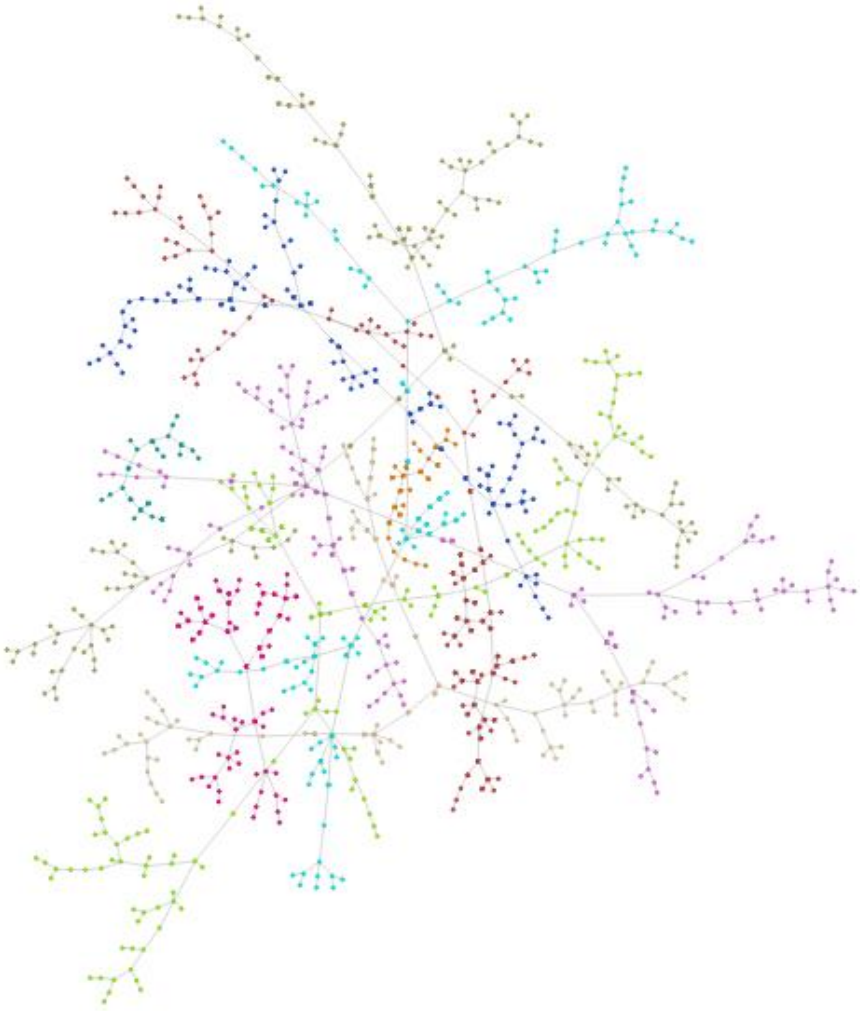


Gambar 8.36 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 7

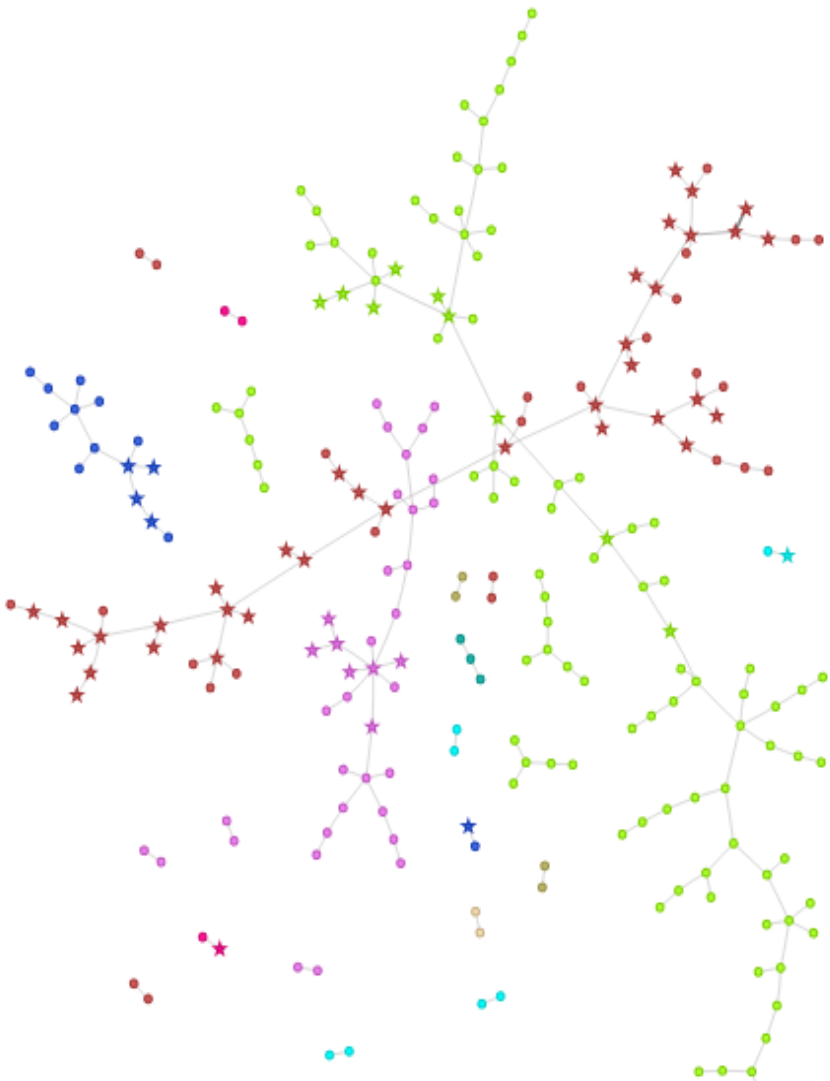


Gambar 8.37 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 7

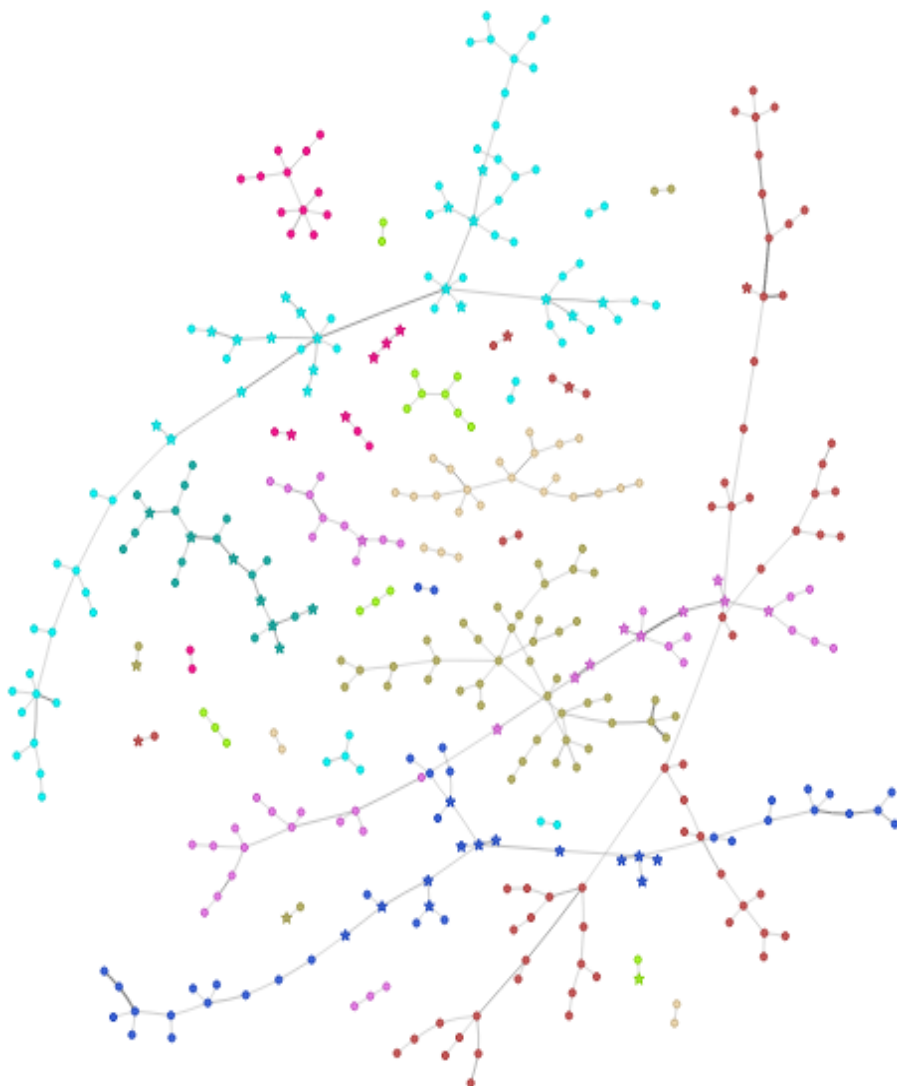
8.1.2 Gambar Graf Cluster Kerjasama Peneliti dengan Nilai K sama dengan 10



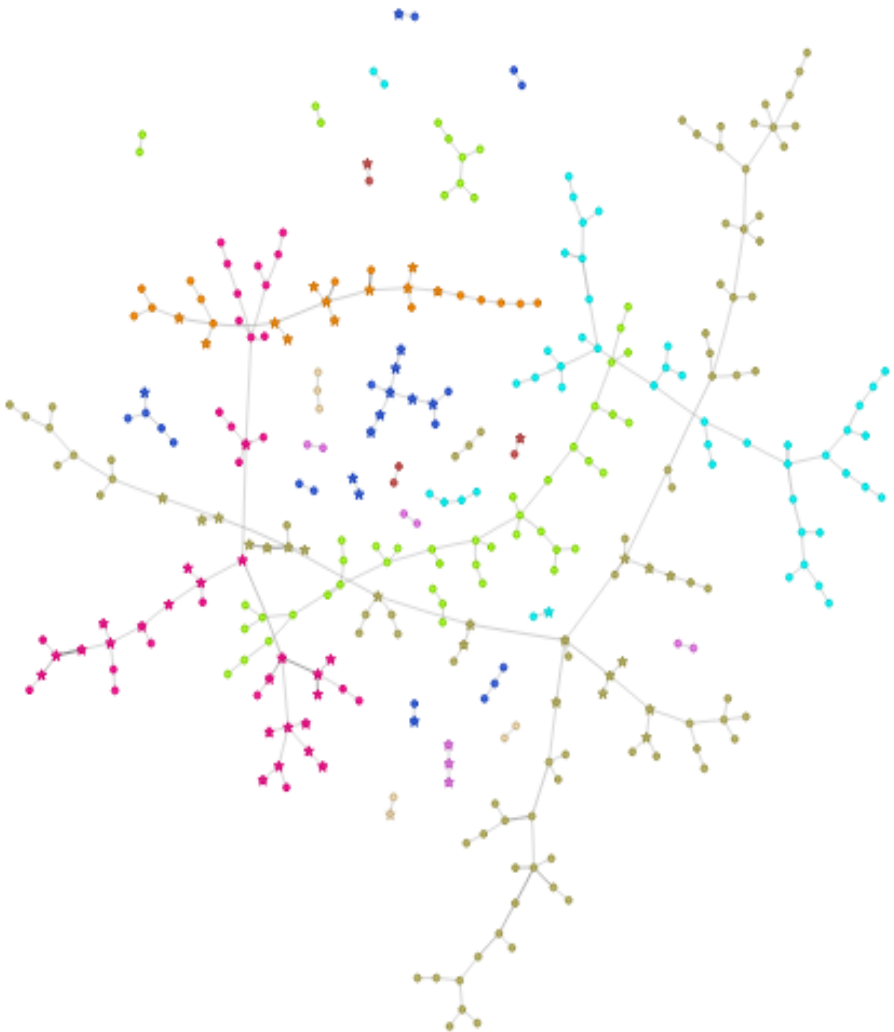
Gambar 8.38 Gambar graf cluster peneliti ITS dengan K sama dengan 10



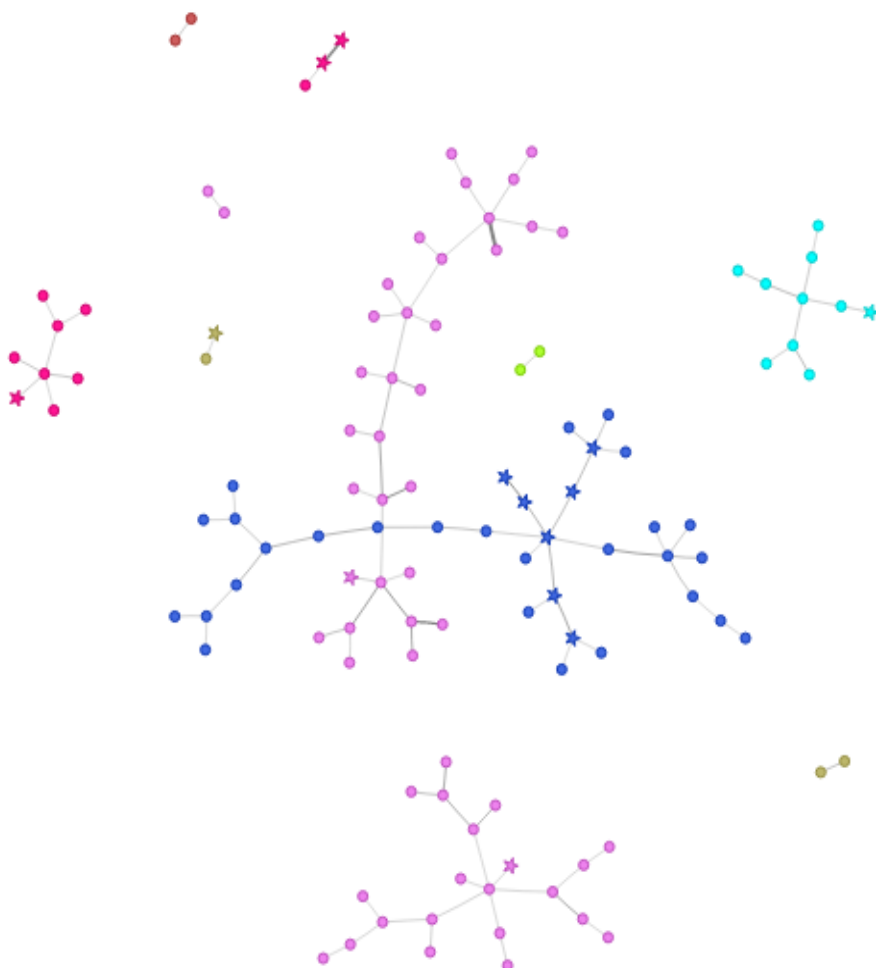
Gambar 8.39 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 10



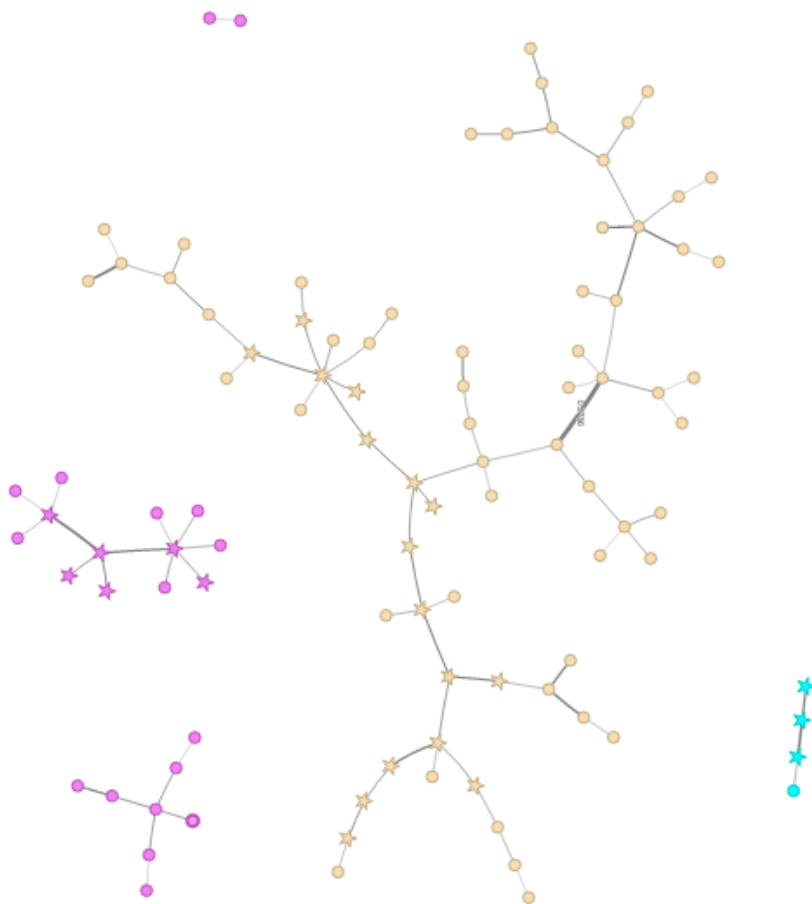
Gambar 8.40 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 10



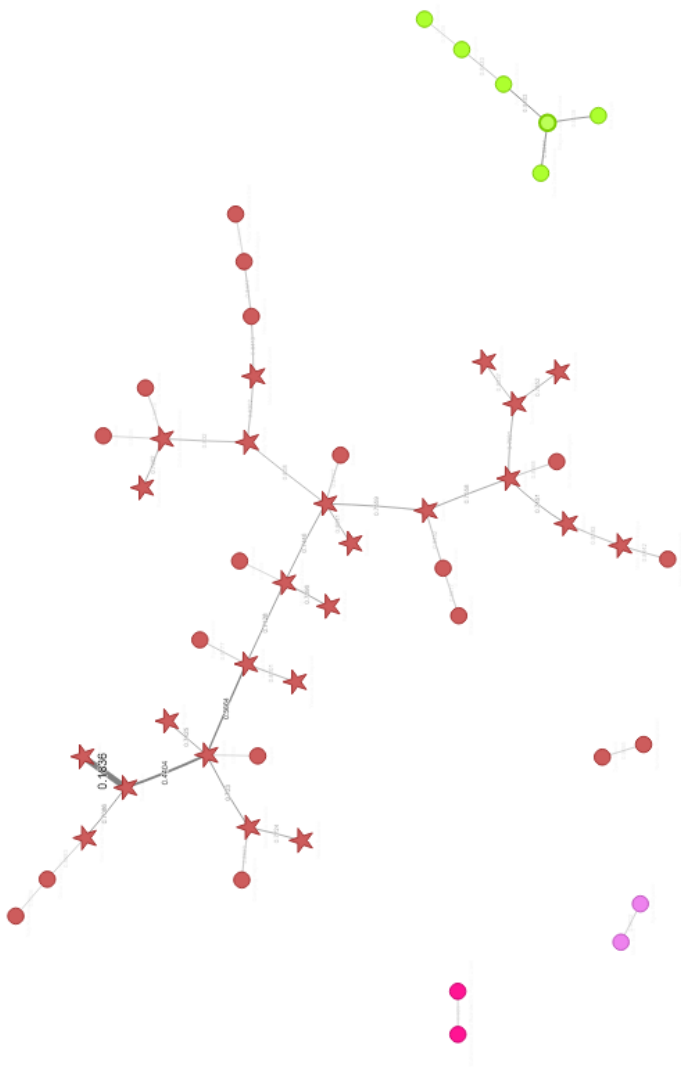
Gambar 8.41 Gambar graf peneliti FTSP ITS dengan K sama dengan 10



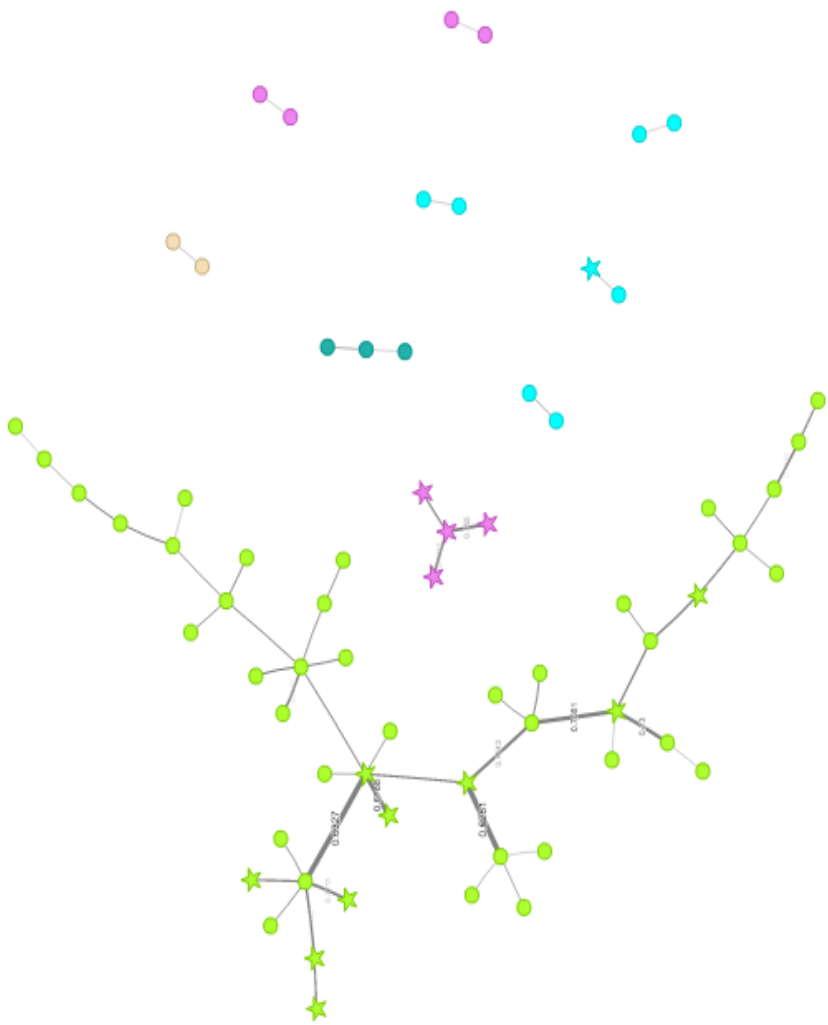
Gambar 8.42 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 10



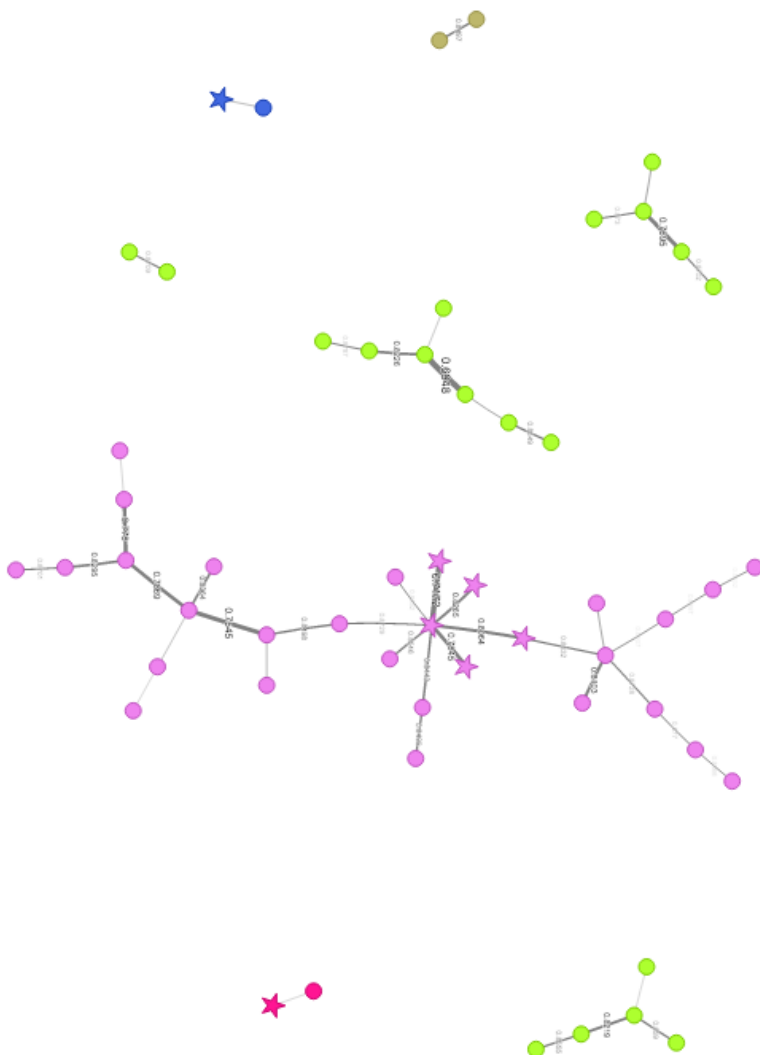
Gambar 8.43 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 10



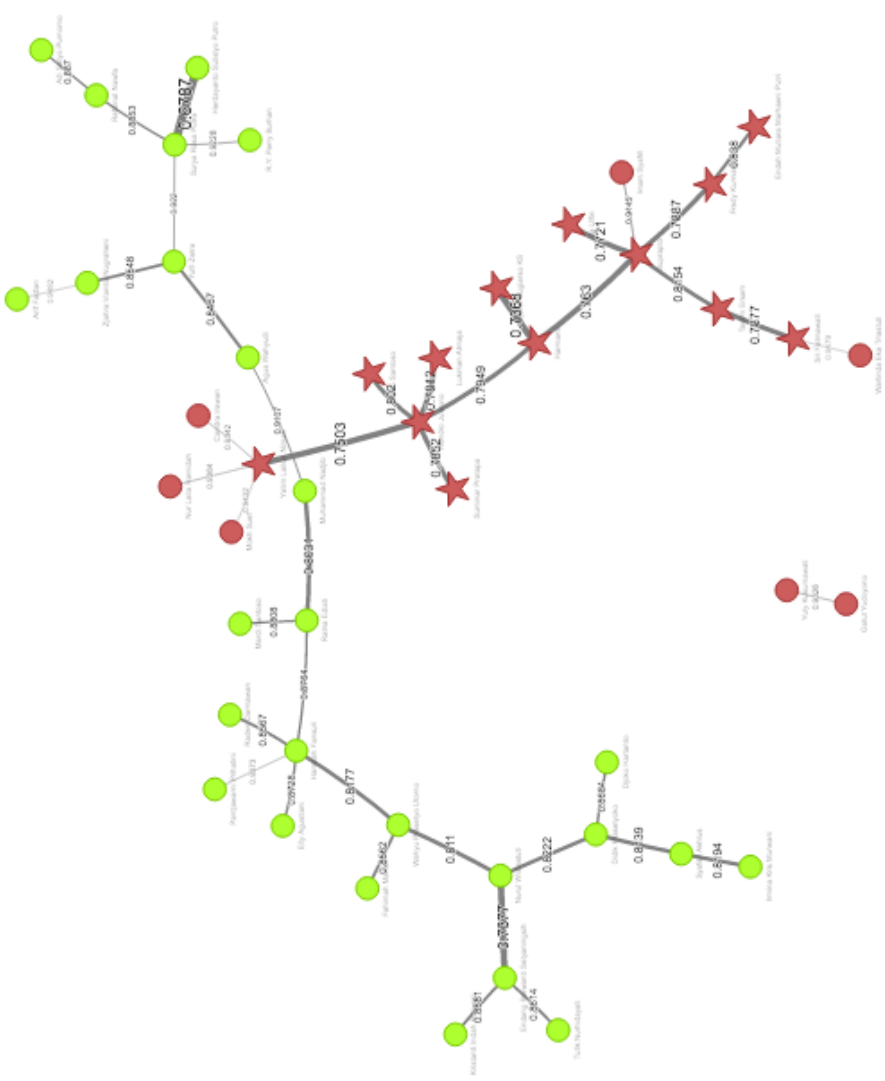
Gambar 8.44 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 10



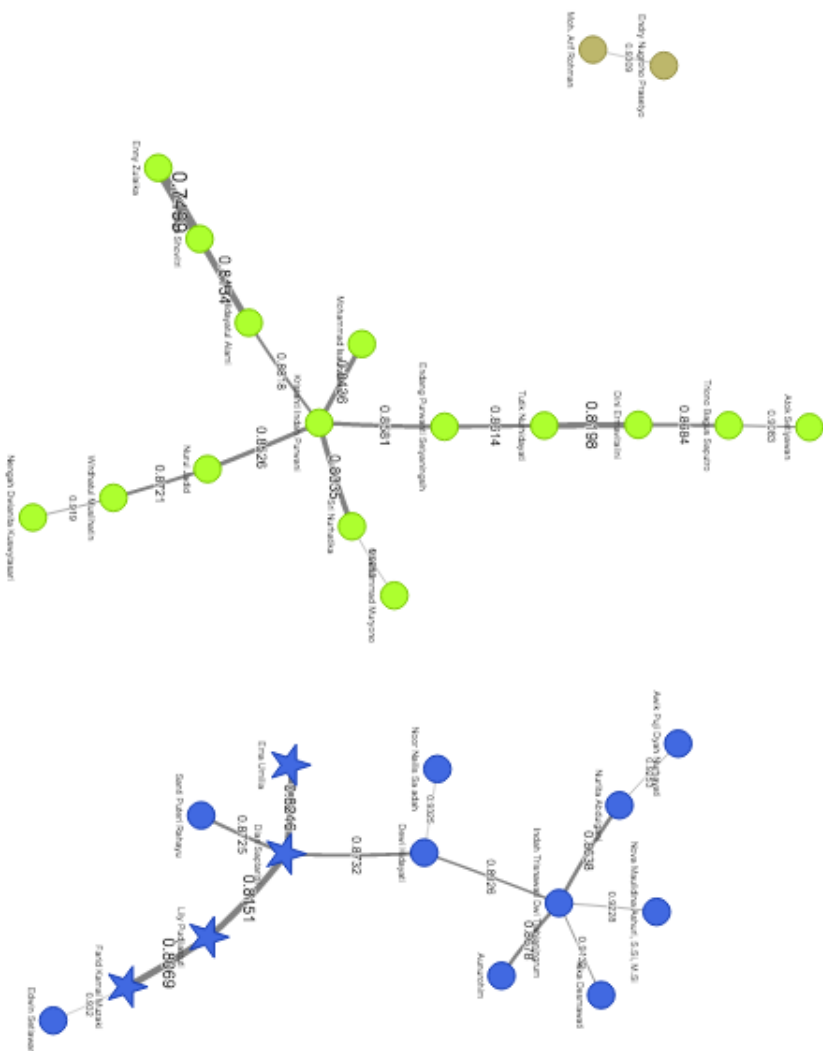
Gambar 8.45 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 10



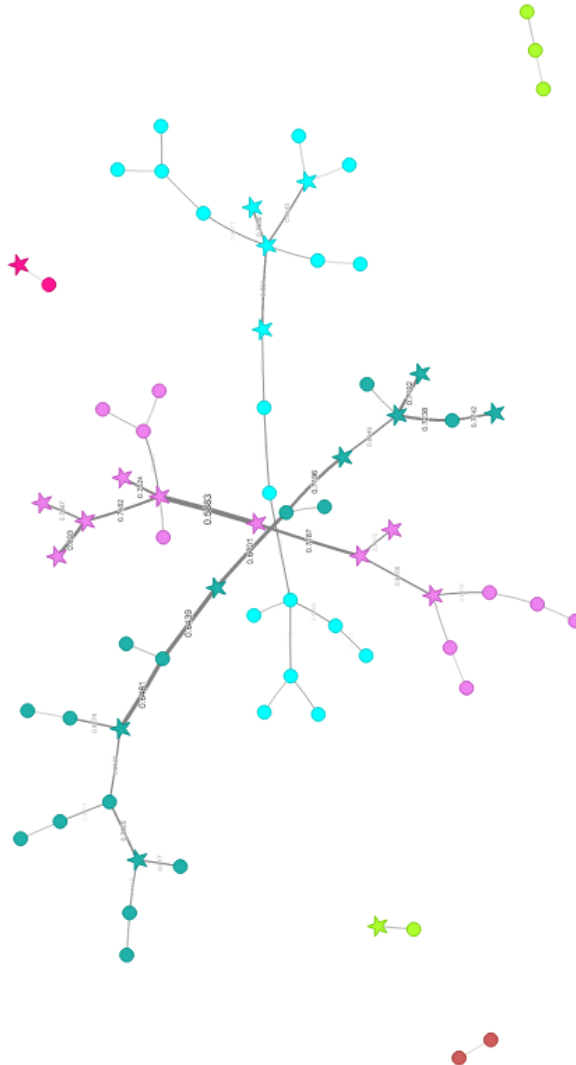
Gambar 8.46 Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 10



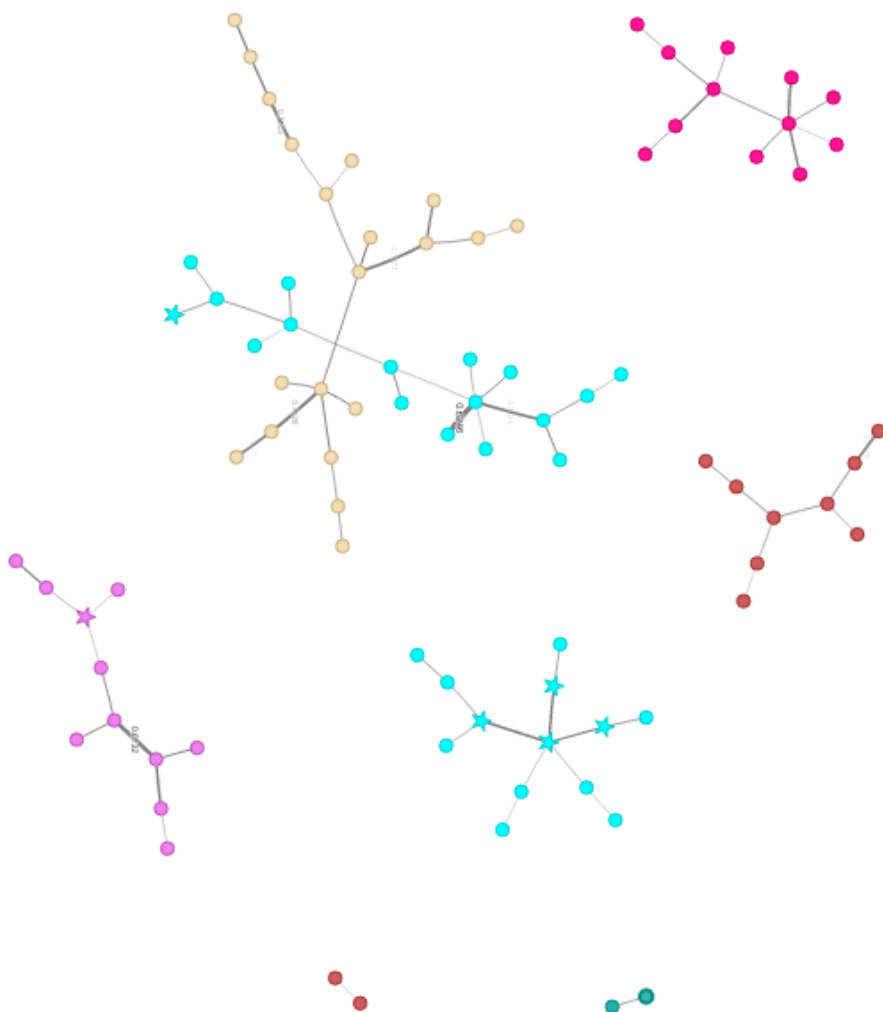
Gambar 8.47 Gambar graf clsuter peneliti jurusan Kimia ITS dengan nilai K sama dengan 10



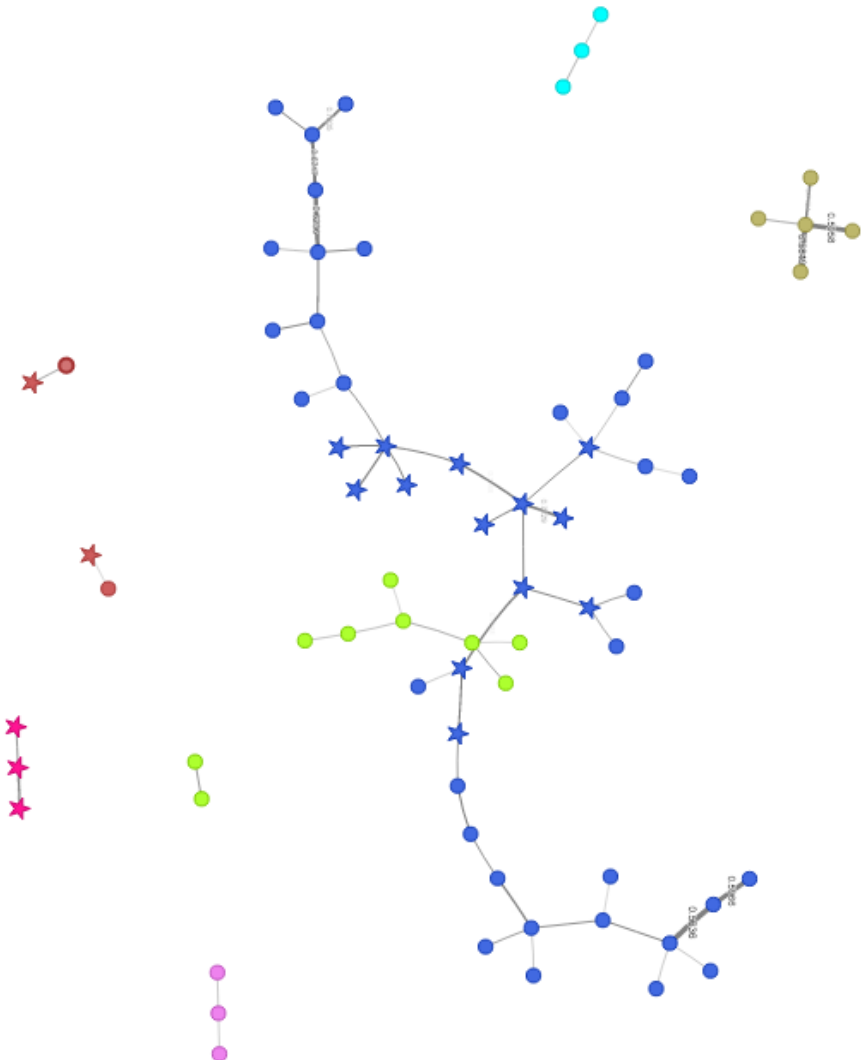
Gambar 8.48 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 10



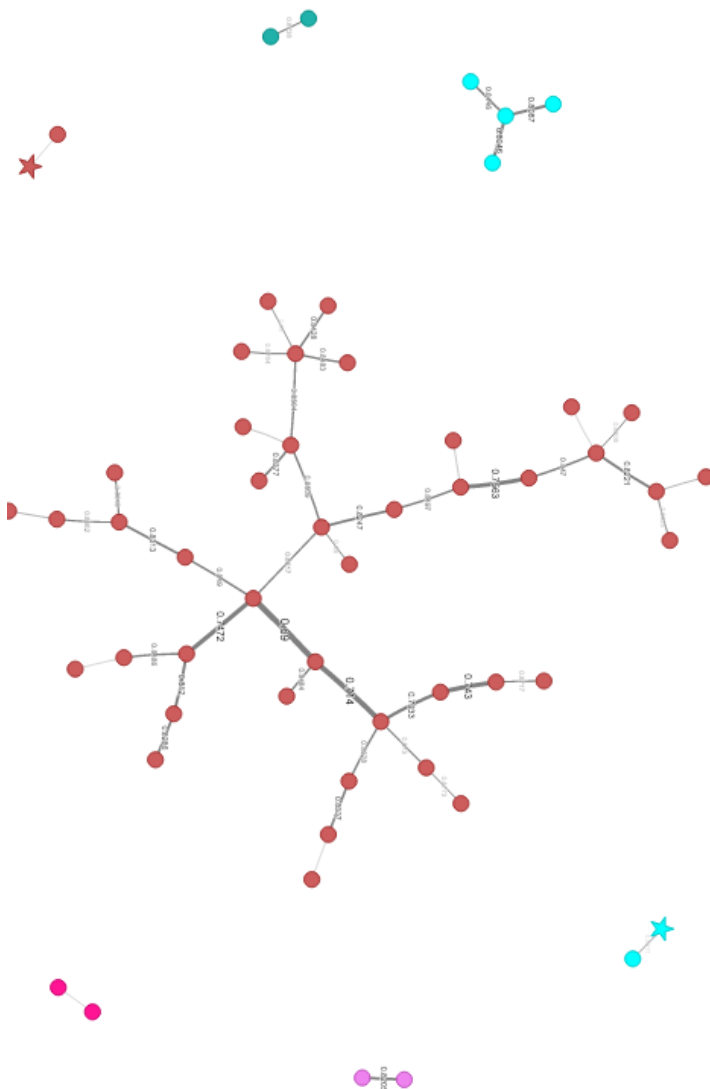
Gambar 8.49 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 10



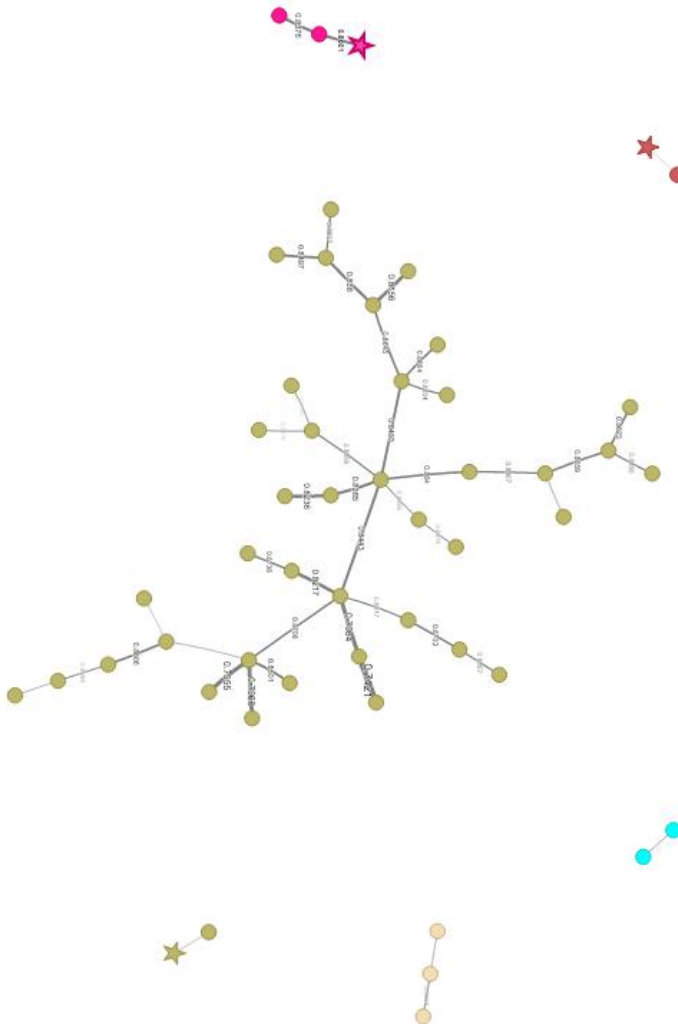
Gambar 8.50 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 10



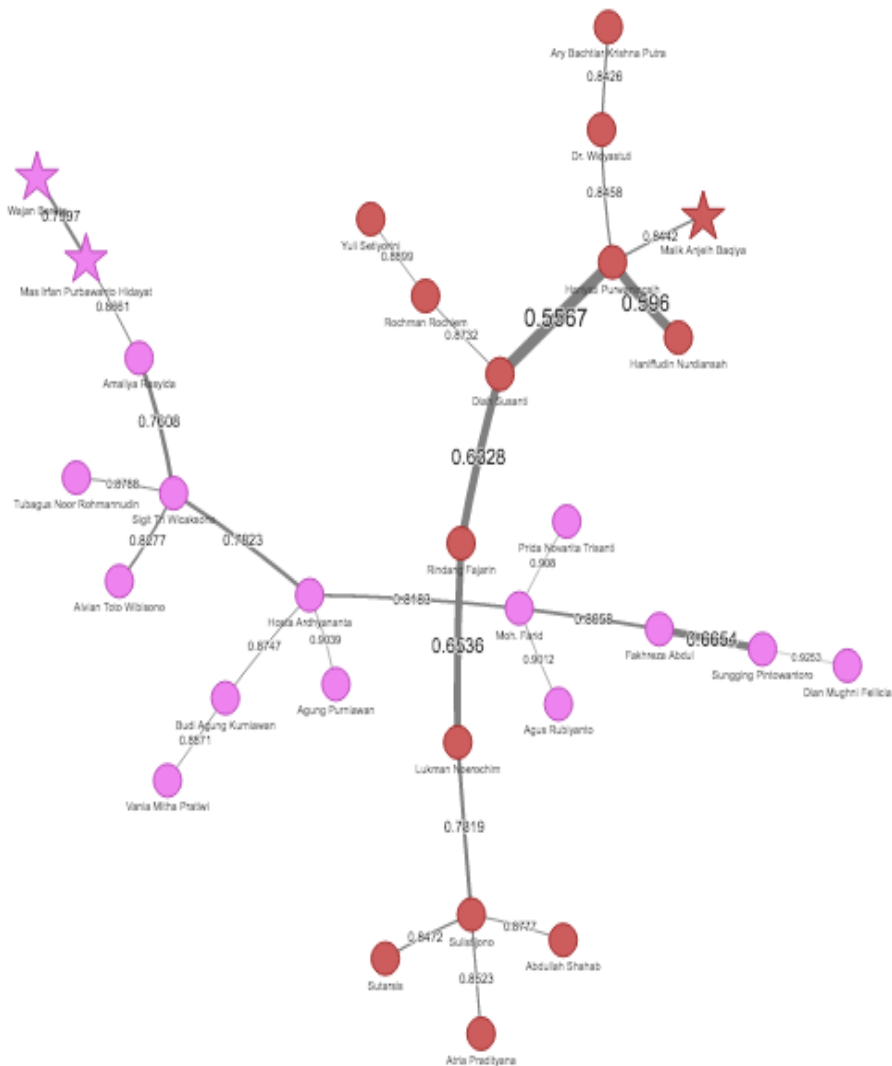
Gambar 8.51 Gambar graf cluster peneliti jurusan Teknik Kimia dengan nilai K sama dengan 10



Gambar 8.52 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 10

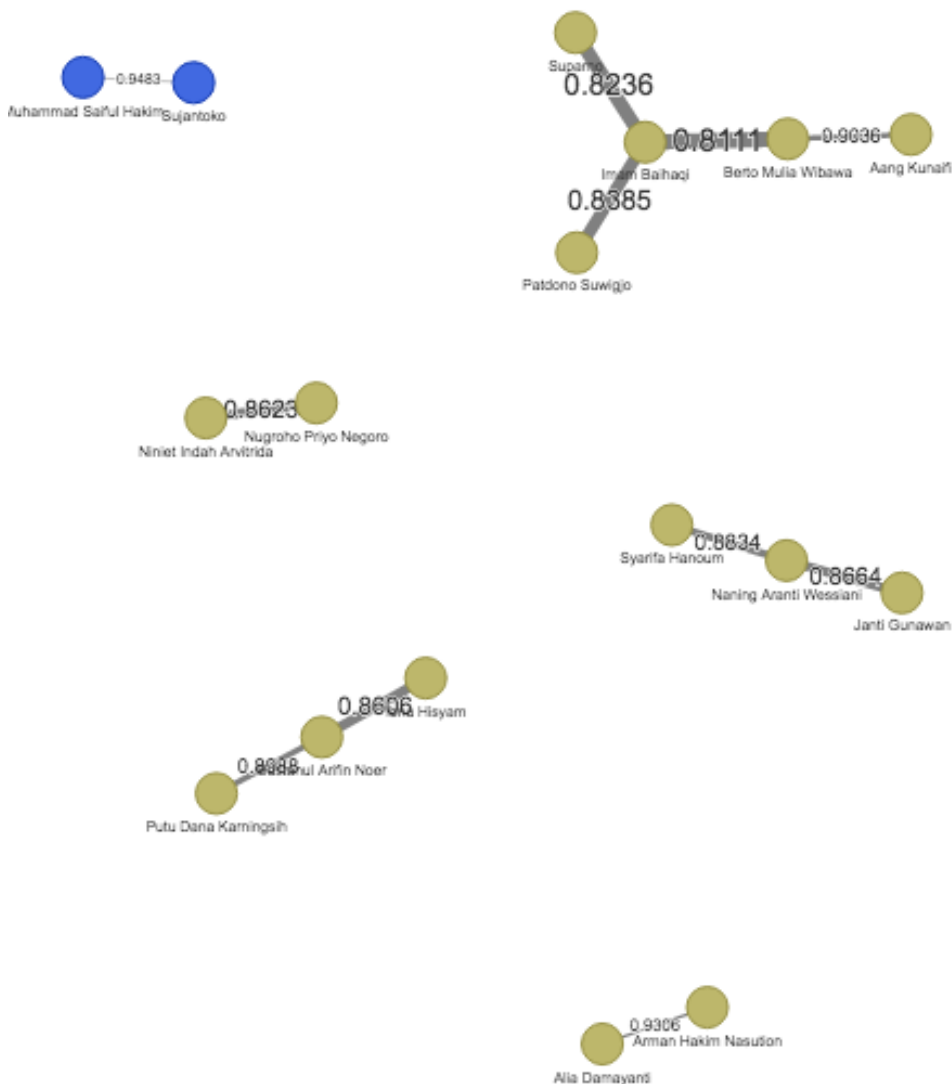


Gambar 8.53 Gambar graf cluster peneliti jurusan Teknik Industri ITS dengan nilai K sama dengan 10

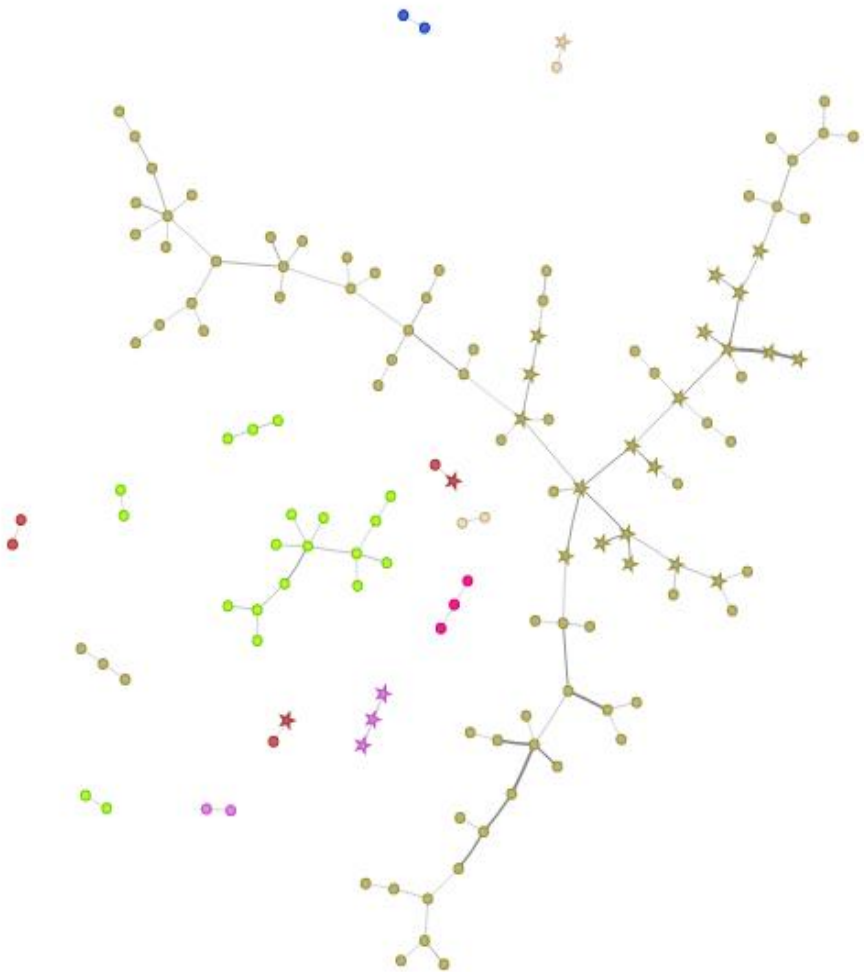


Gambar 8.54 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 10

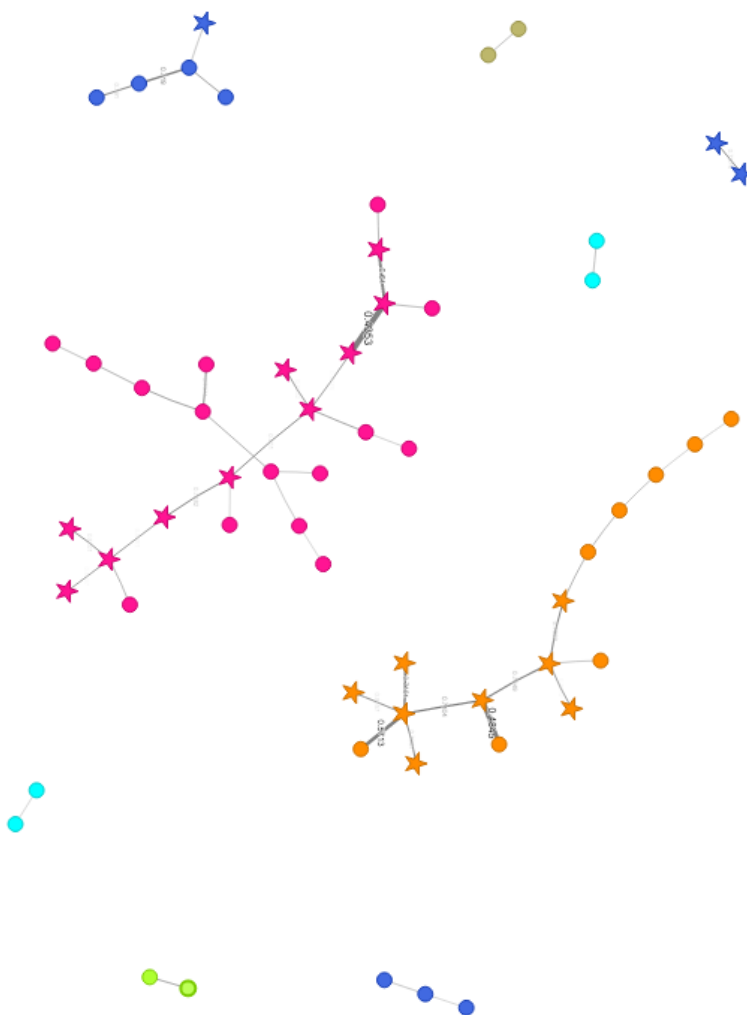
Gambar 8.55 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 10



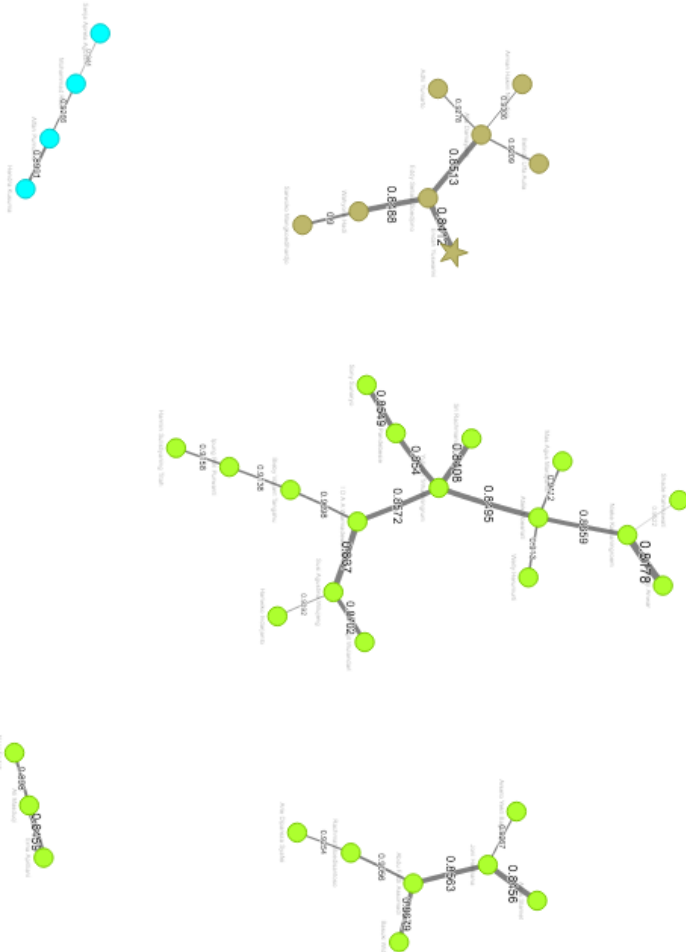
Gambar 8.56 Gambar graf cluster peneliti jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 10



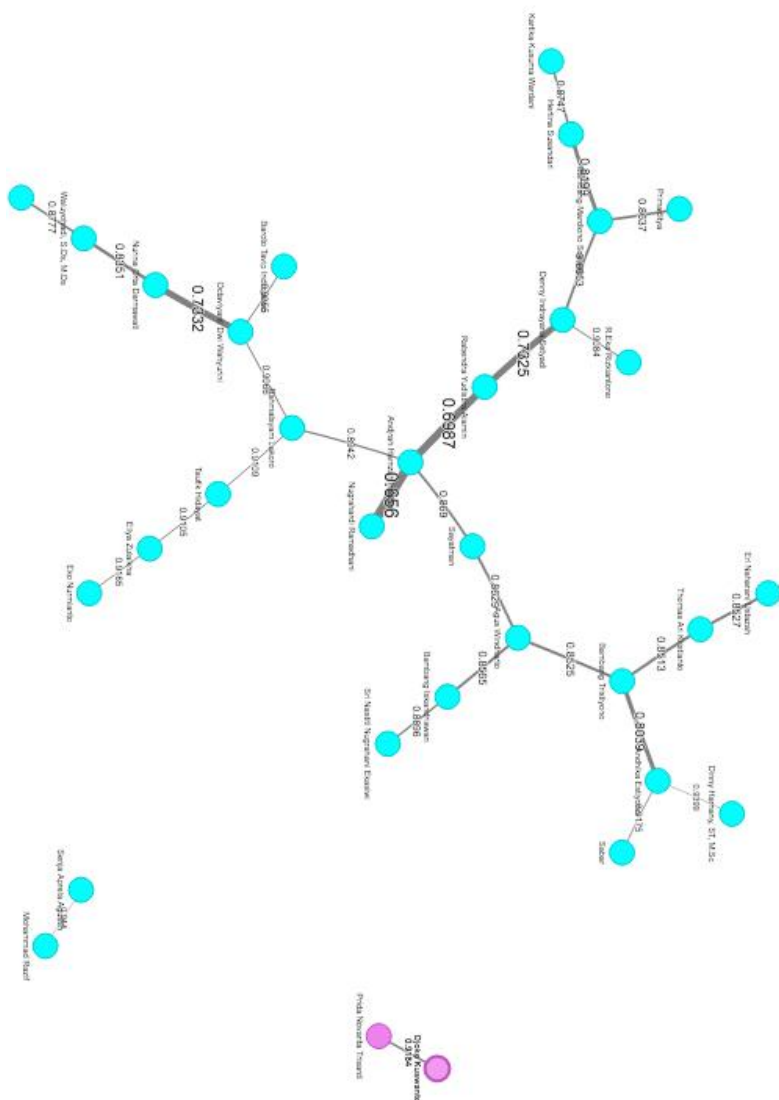
Gambar 8.57 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 10



Gambar 8.58 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 10

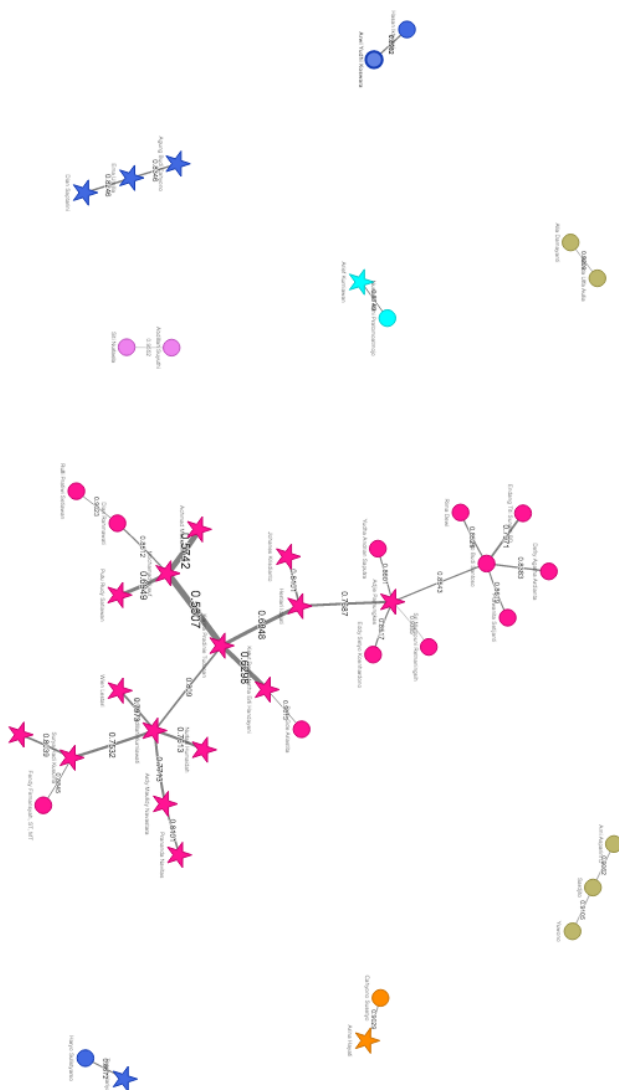


Gambar 8.59 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 10

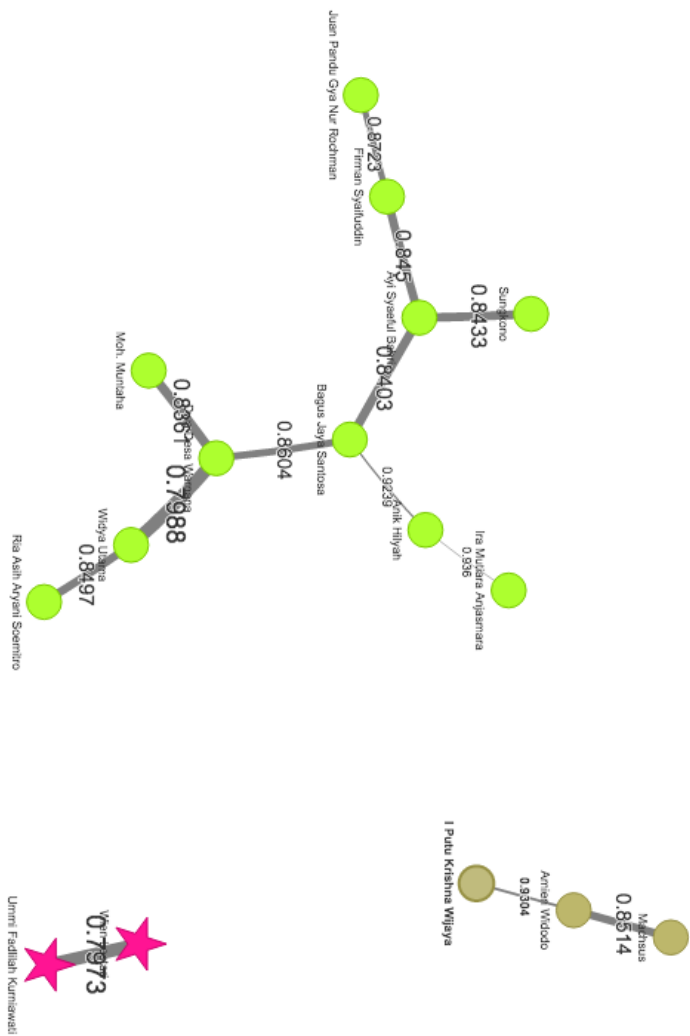


Gambar 8.60 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 10

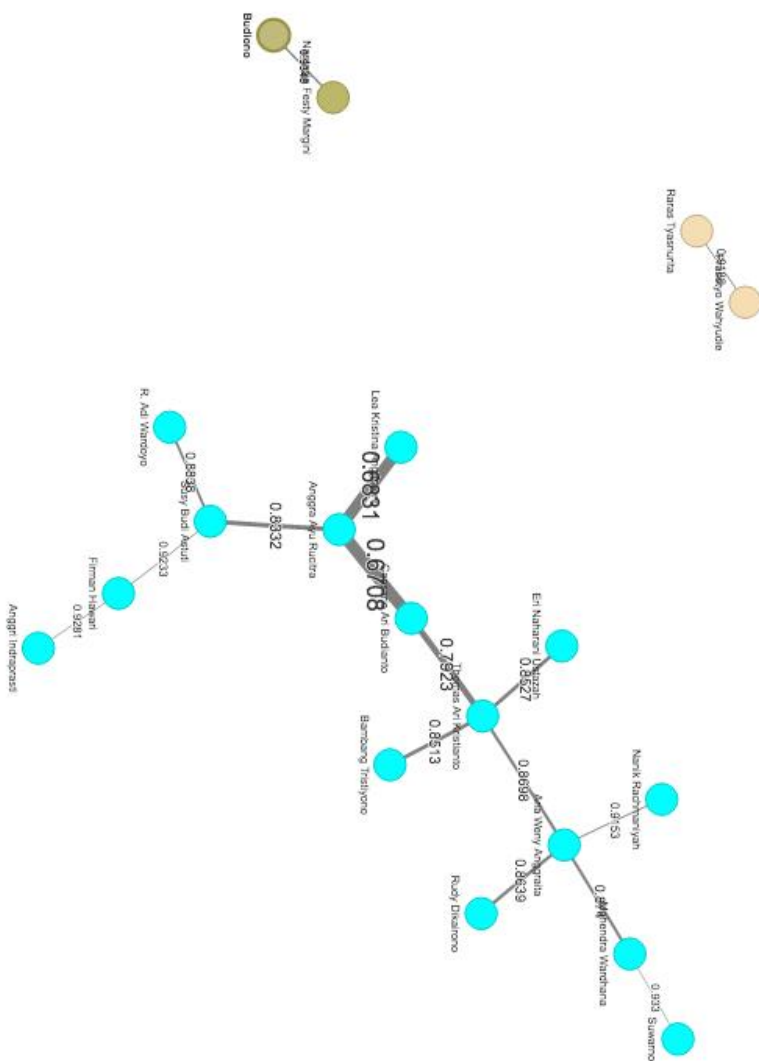
Gambar 8.61 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 10



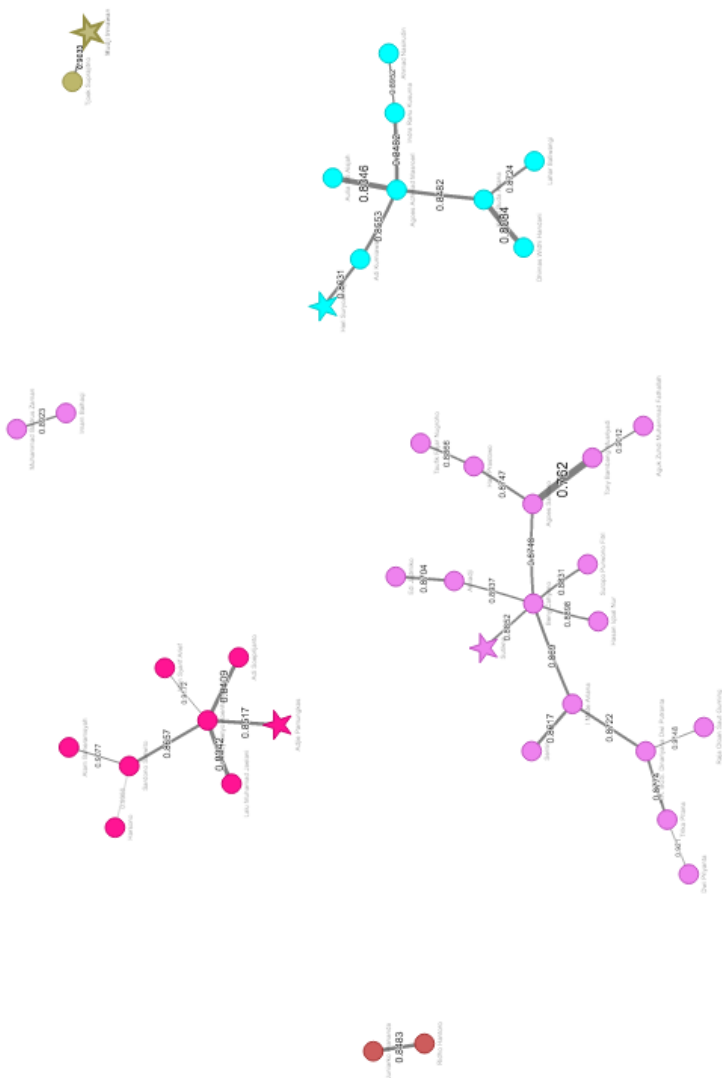
Gambar 8.62 Gambar graf cluster peneliti jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 10



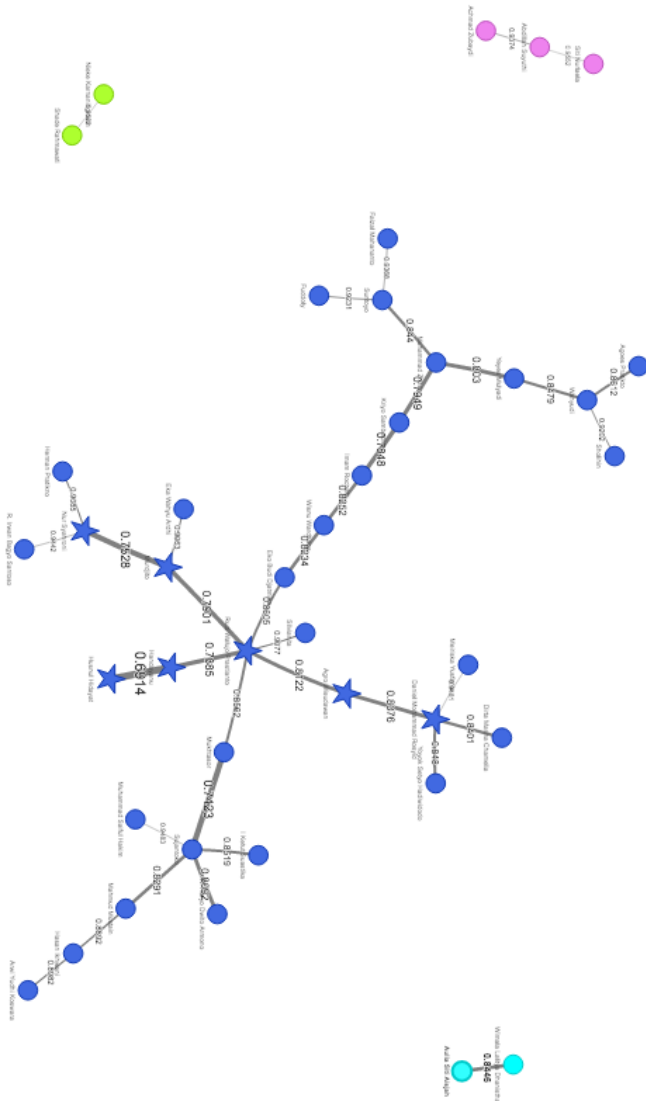
Gambar 8.63 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 10



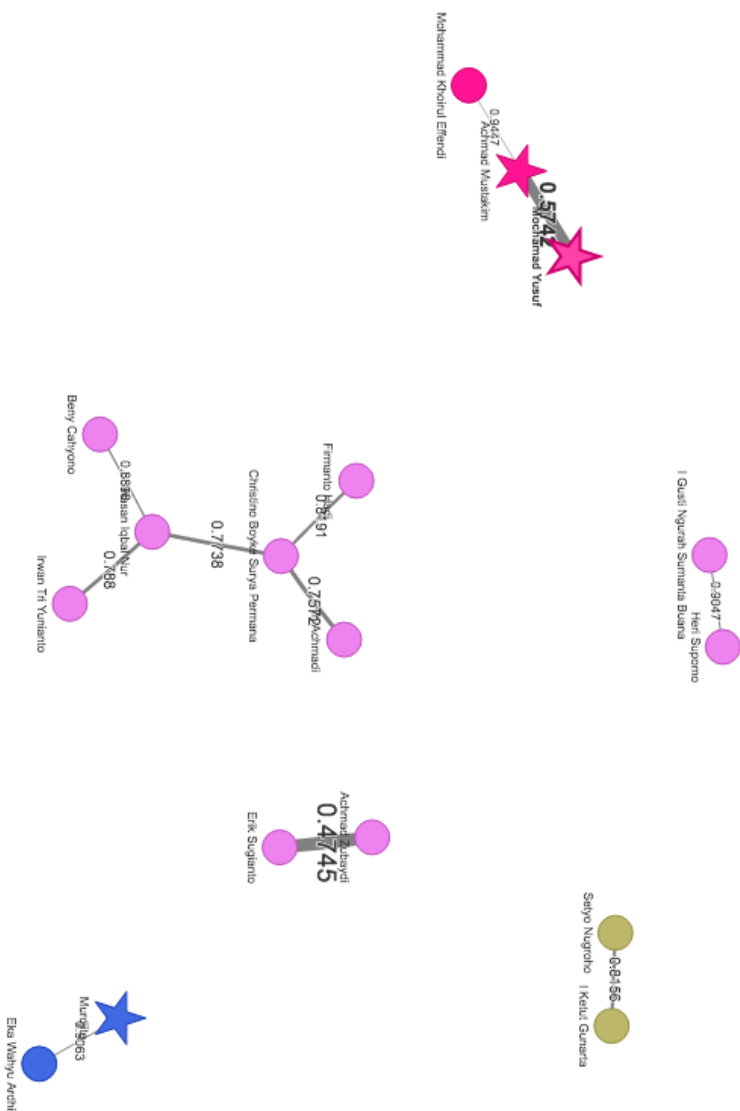
Gambar 8.64 Gambar graf cluster peneliti jurusan Desain Interior ITS dengan nilai K sama dengan 10



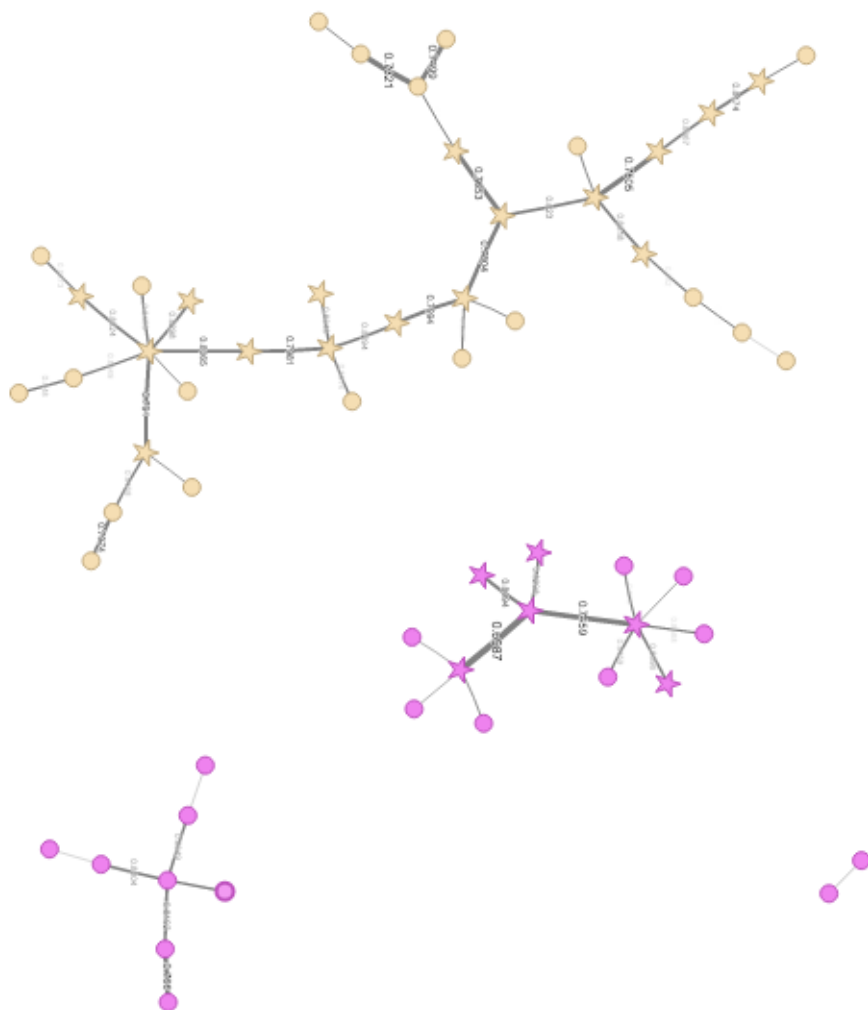
Gambar 8.66 Gambar graf cluster peneliti jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 10



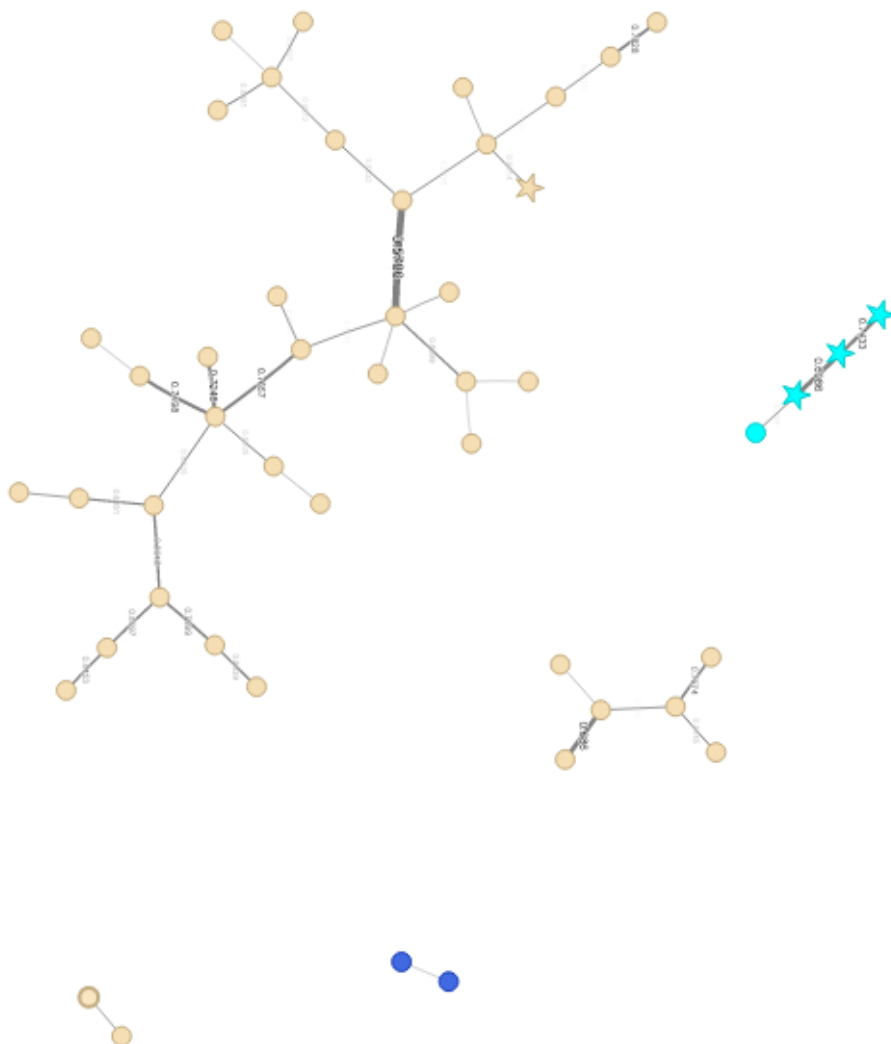
Gambar 8.67 Gambar graf cluster peneliti jurusan Teknik Kelautan ITS dengan nilai K sama dengan 10



Gambar 8.68 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 10

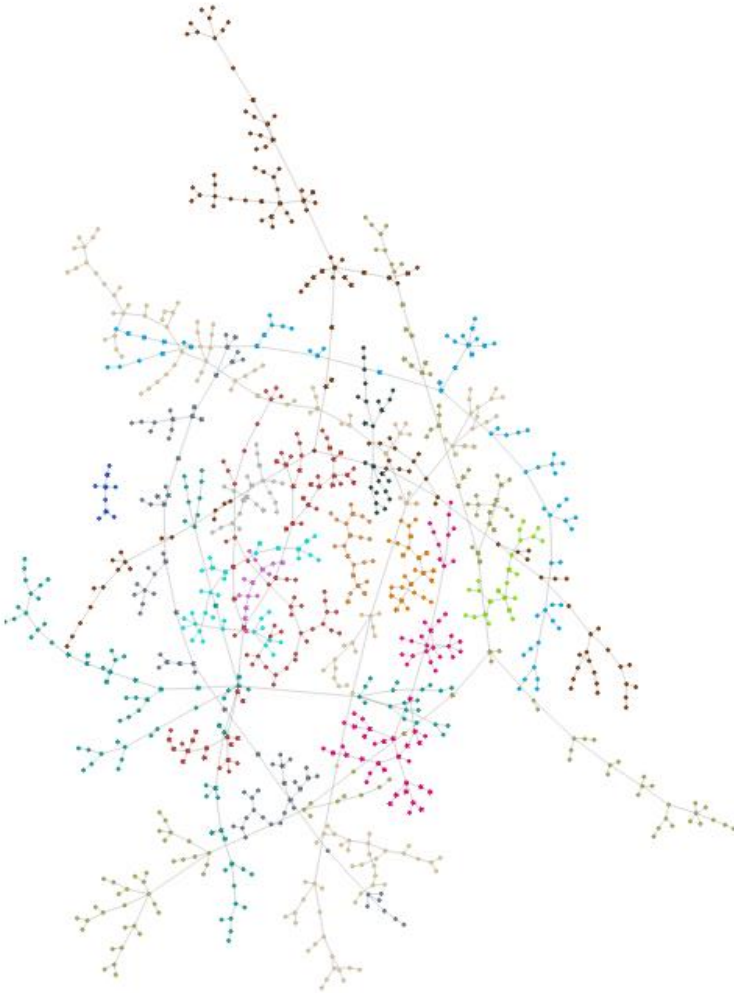


Gambar 8.69 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 10

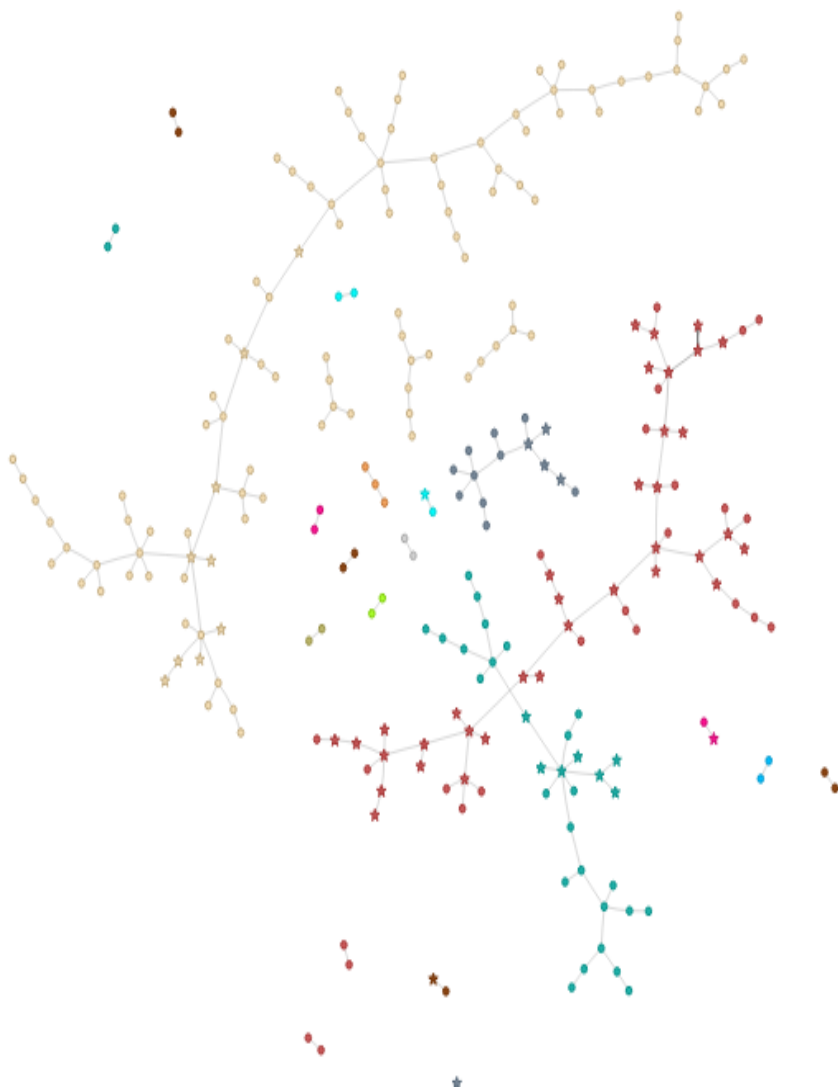


Gambar 8.70 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 10

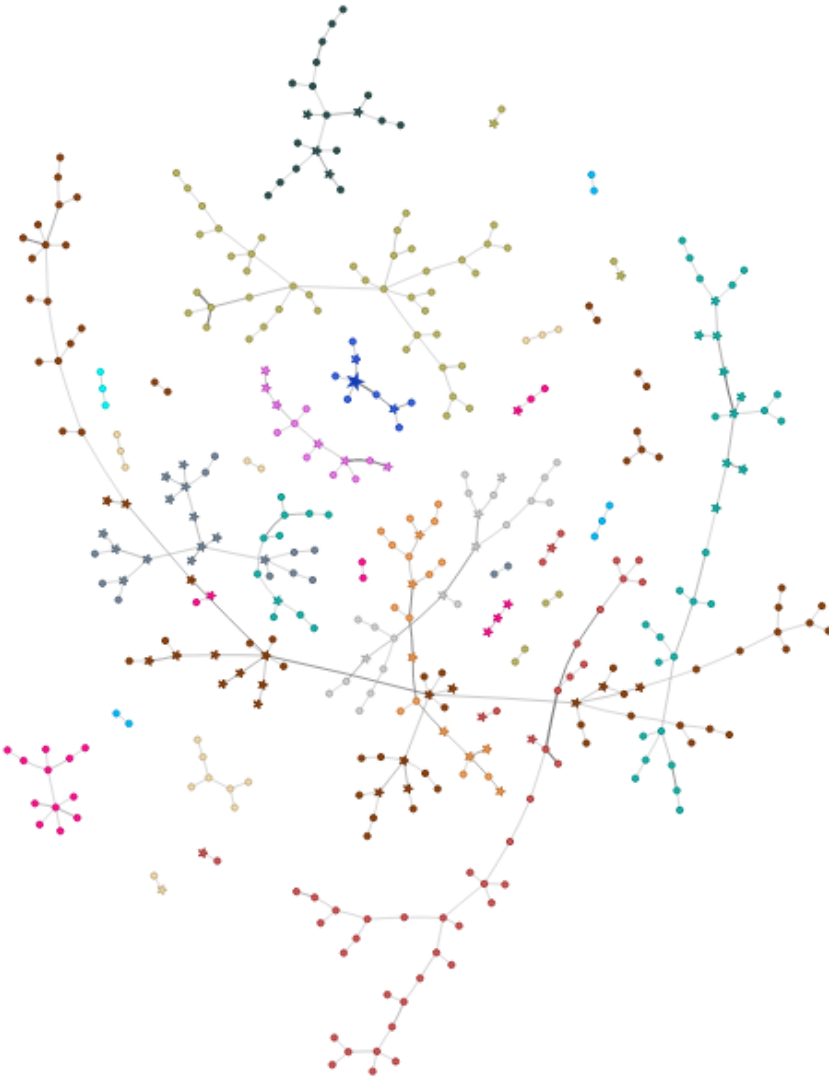
8.1.3 Gambar Graf Cluster Kerjasama Peneliti dengan Nilai K sama dengan 16



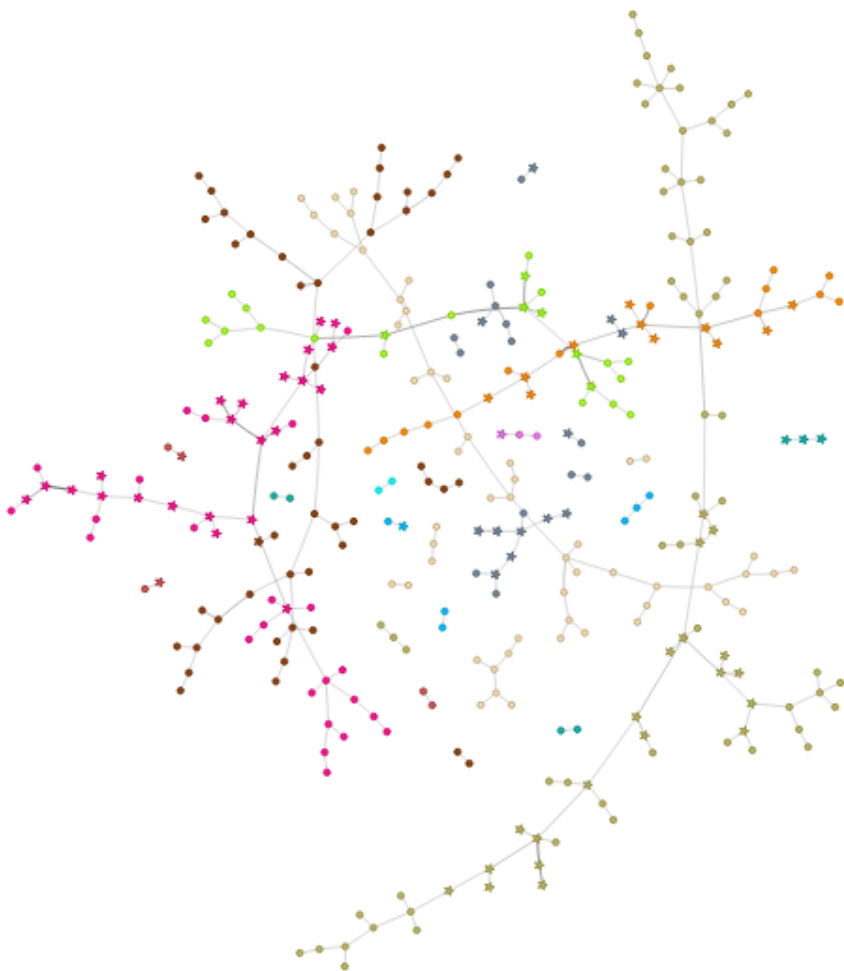
Gambar 8.71 Gambar graf cluster peneliti ITS dengan K sama dengan 16



Gambar 8.72 Gambar graf cluster peneliti FMIPA ITS dengan K sama dengan 16



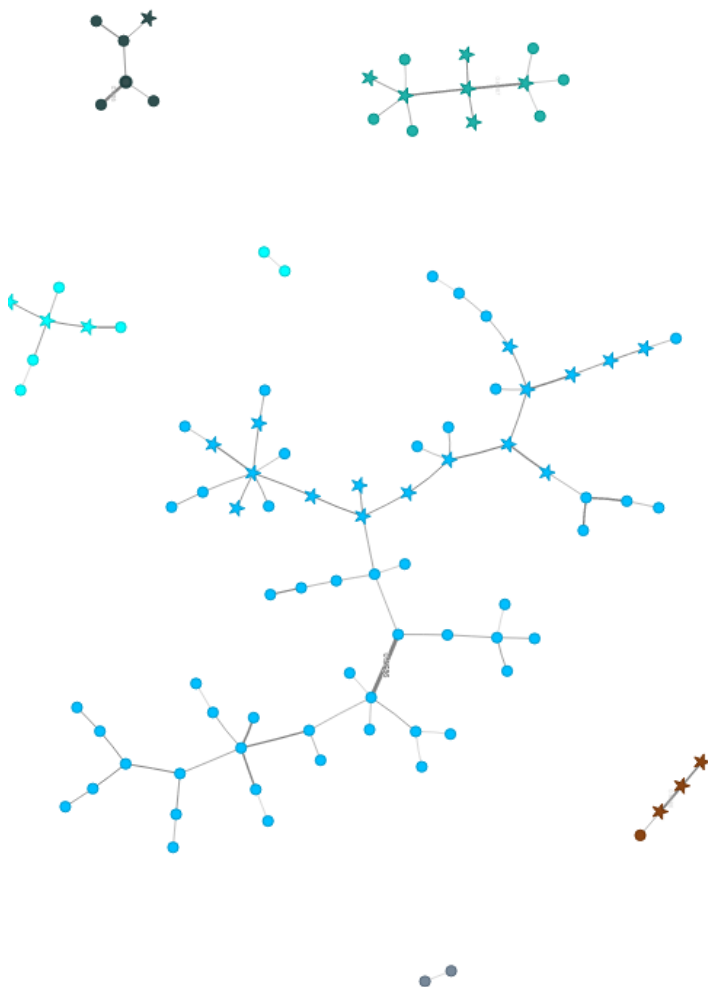
Gambar 8.73 Gambar graf cluster peneliti FTI ITS dengan K sama dengan 16



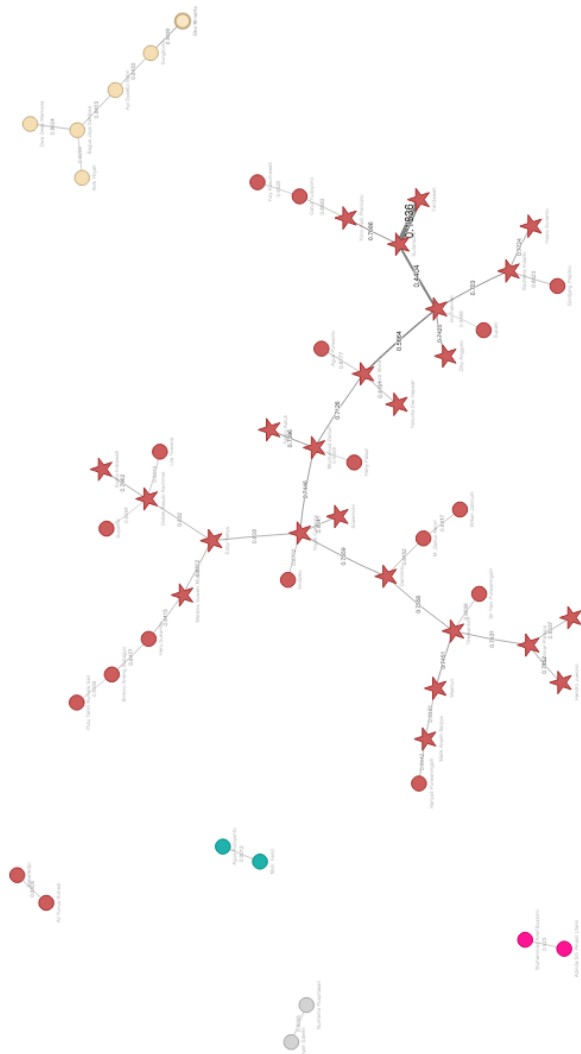
Gambar 8.74 Gambar graf peneliti FTSP ITS dengan K sama dengan 16



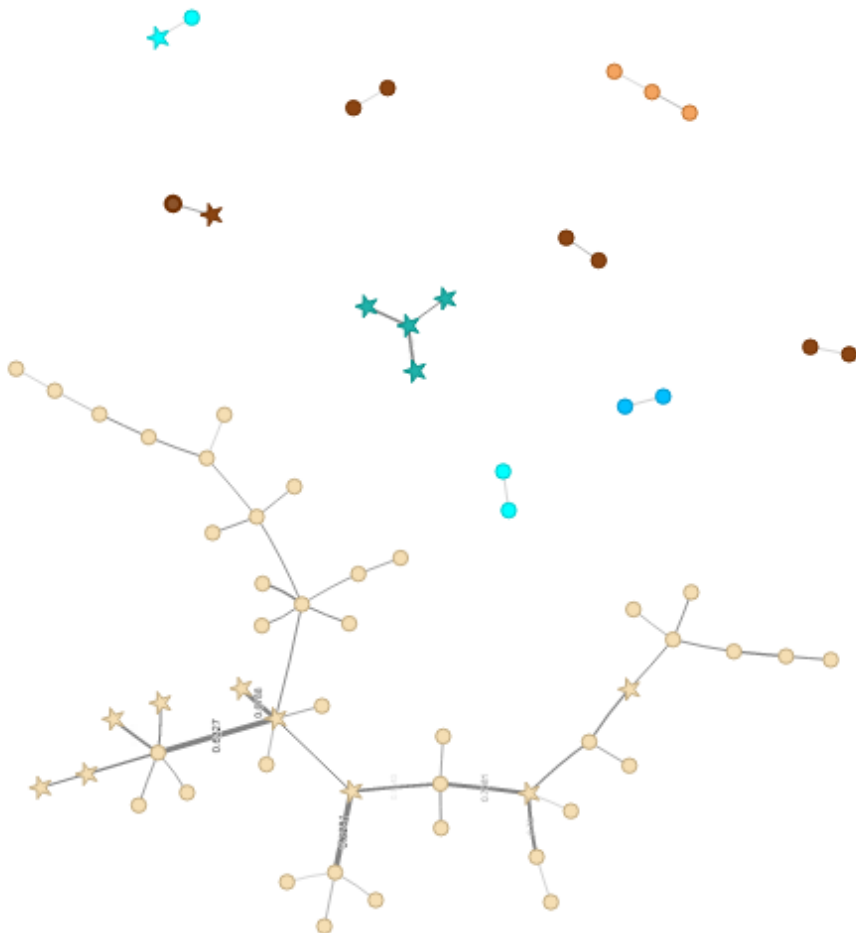
Gambar 8.75 Gambar graf cluster peneliti FTK ITS dengan nilai K sama dengan 16



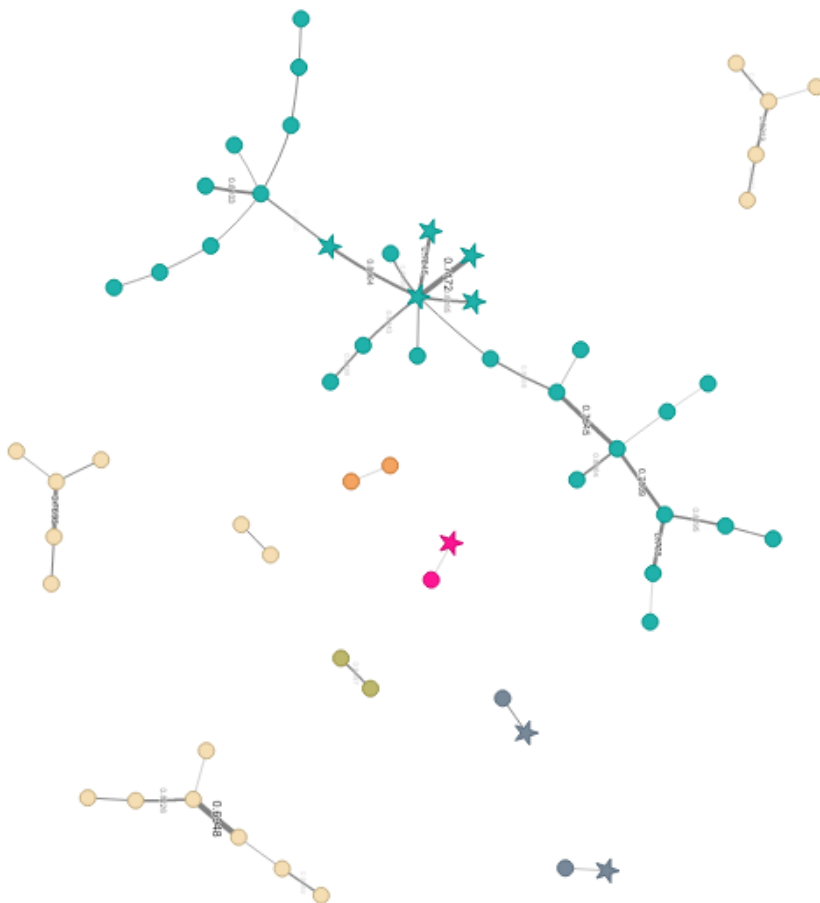
Gambar 8.76 Gambar graf cluster peneliti FTIF ITS dengan nilai K sama dengan 16



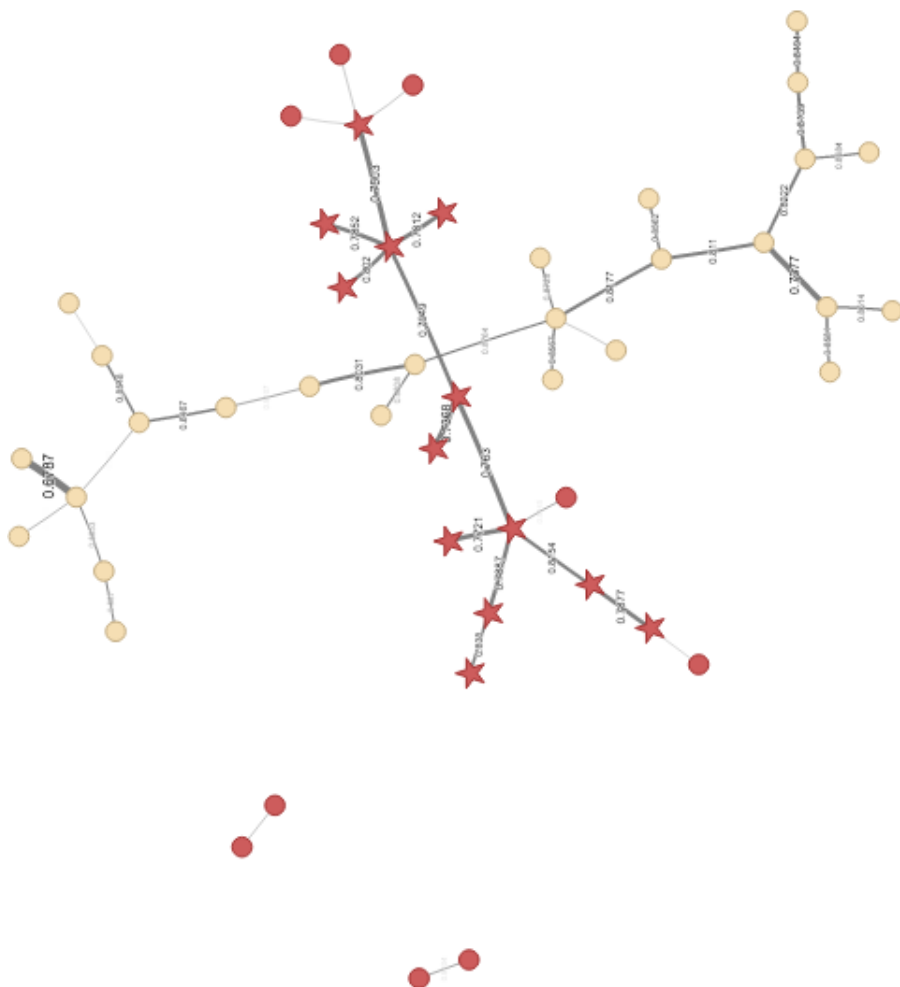
Gambar 8.77 Gambar graf cluster peneliti jurusan Fisika ITS dengan nilai K sama dengan 16



Gambar 8.78 Gambar graf cluster peneliti jurusan Matematika ITS dengan nilai K sama dengan 16

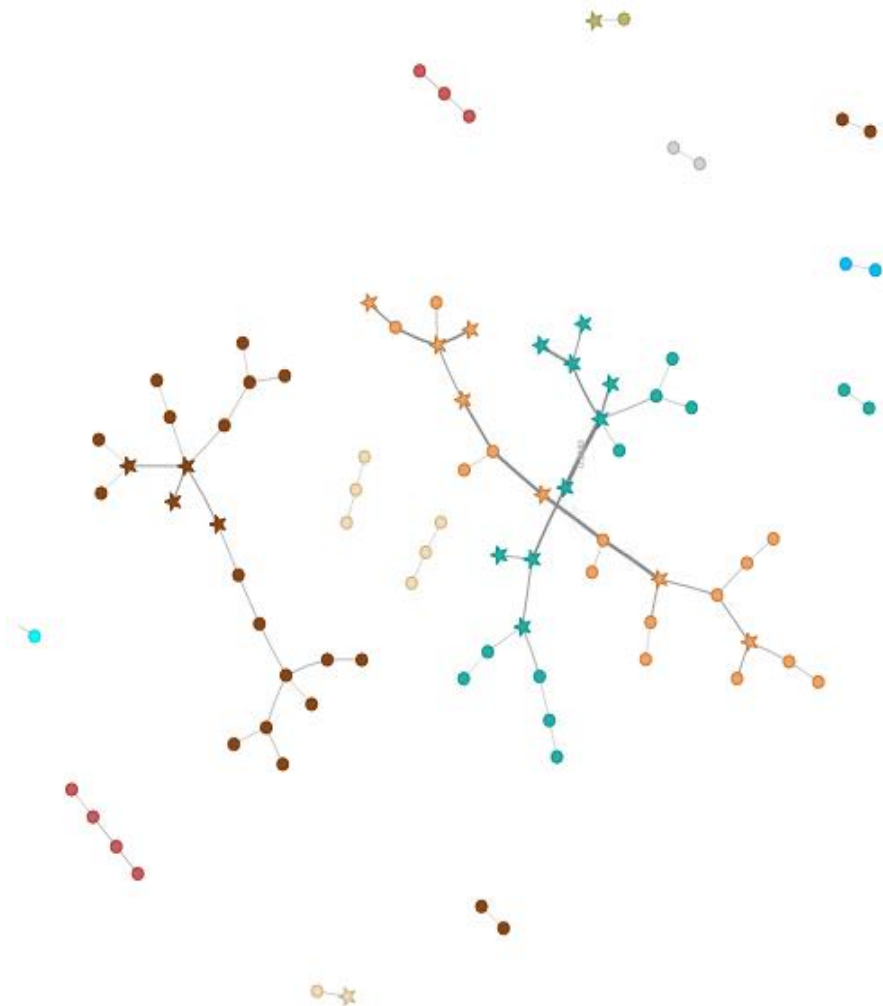


Gambar 8.79 Gambar graf cluster peneliti jurusan Statistika ITS dengan nilai K sama dengan 16

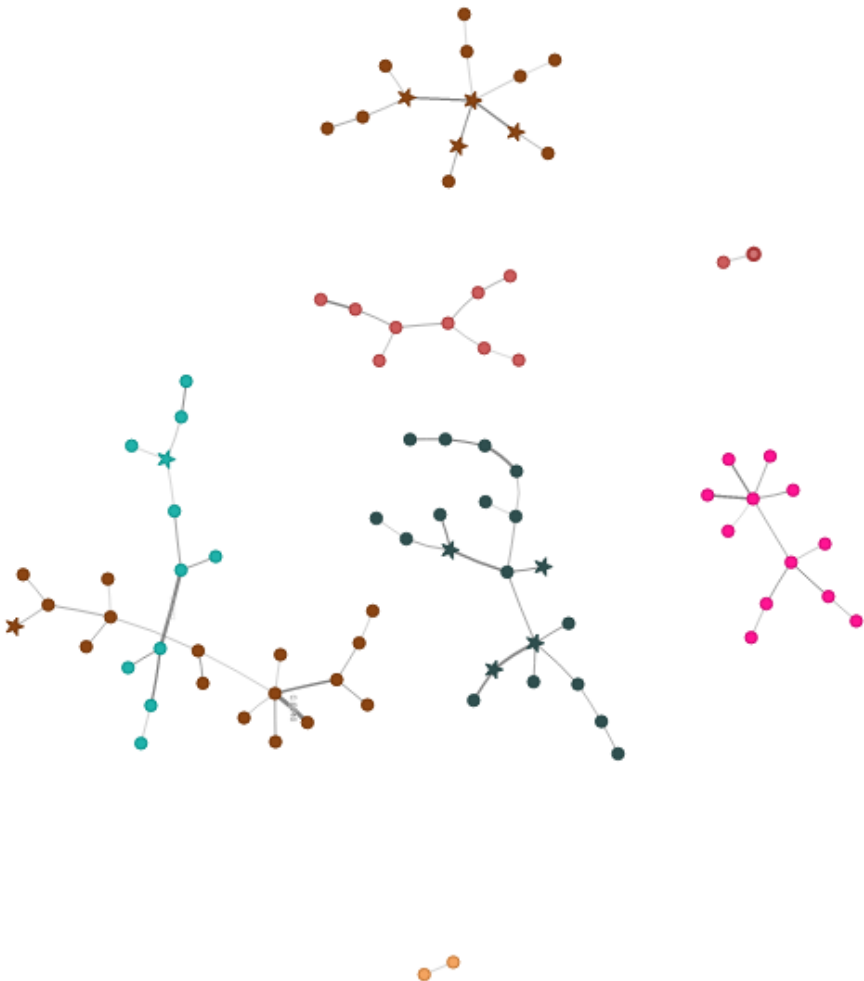


Gambar 8.80 Gambar graf cluster peneliti jurusan Kimia ITS dengan nilai K sama dengan 16

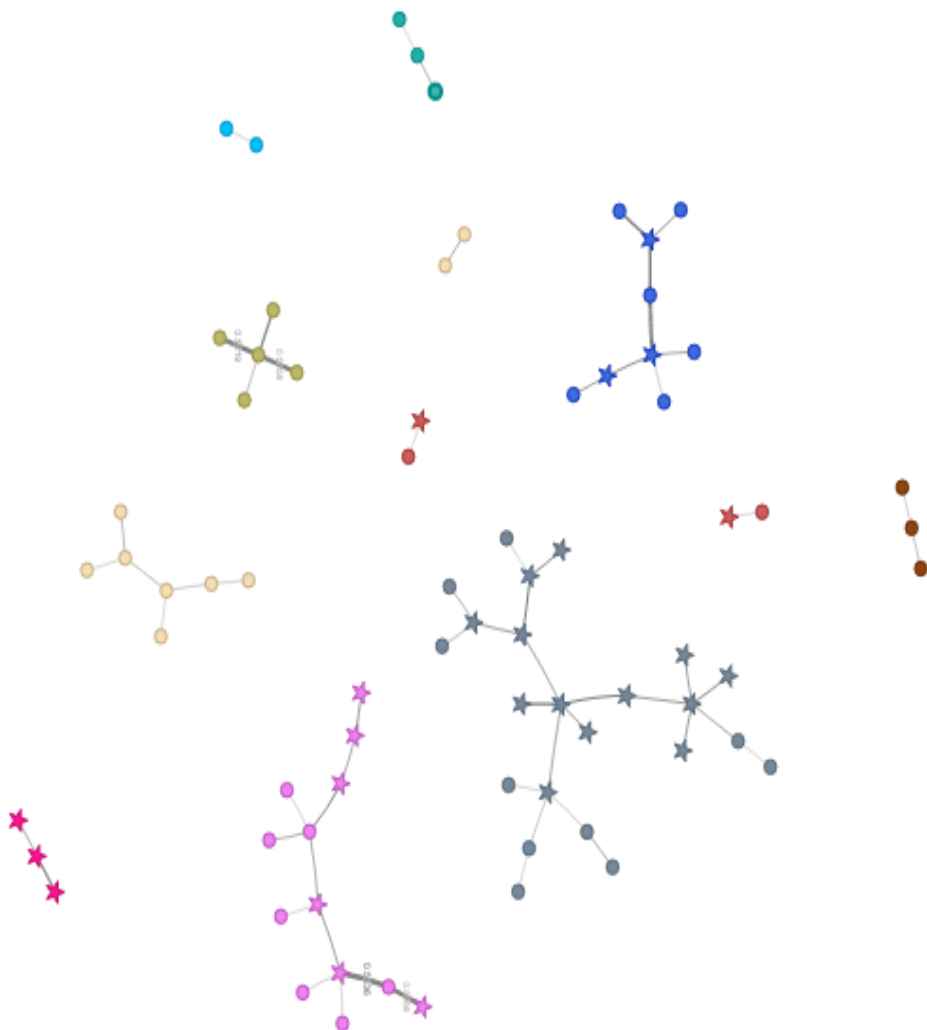
Gambar 8.81 Gambar graf cluster peneliti jurusan Biologi ITS dengan nilai K sama dengan 16



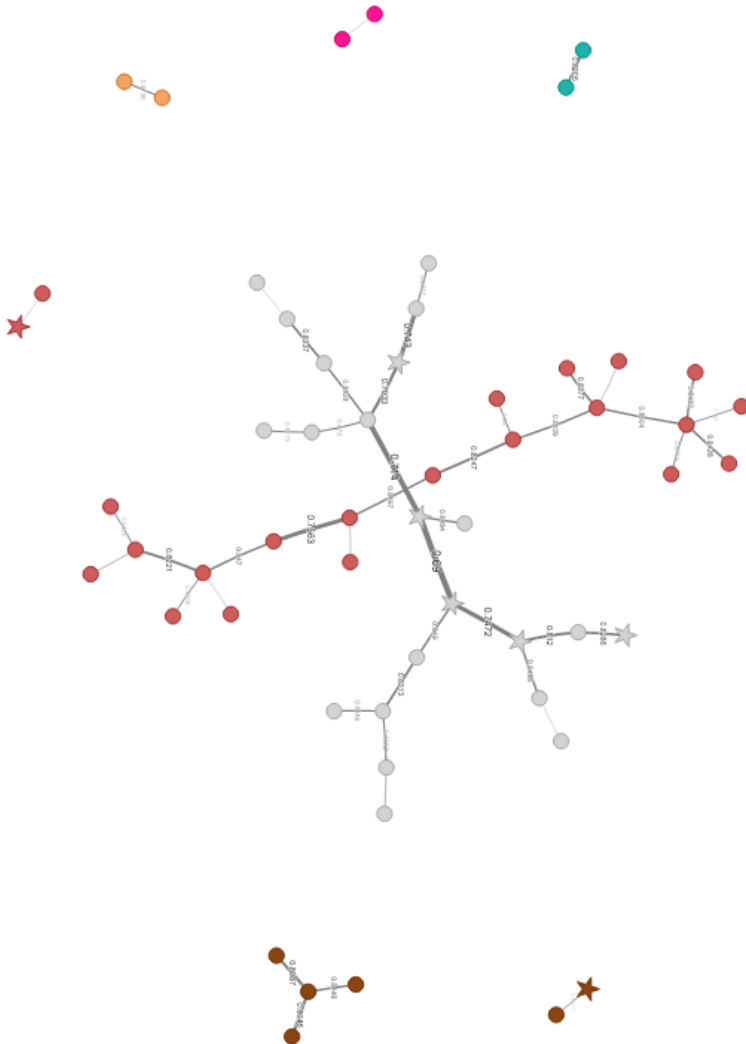
Gambar 8.82 Gambar graf cluster peneliti jurusan Teknik Mesin ITS dengan nilai K sama dengan 16



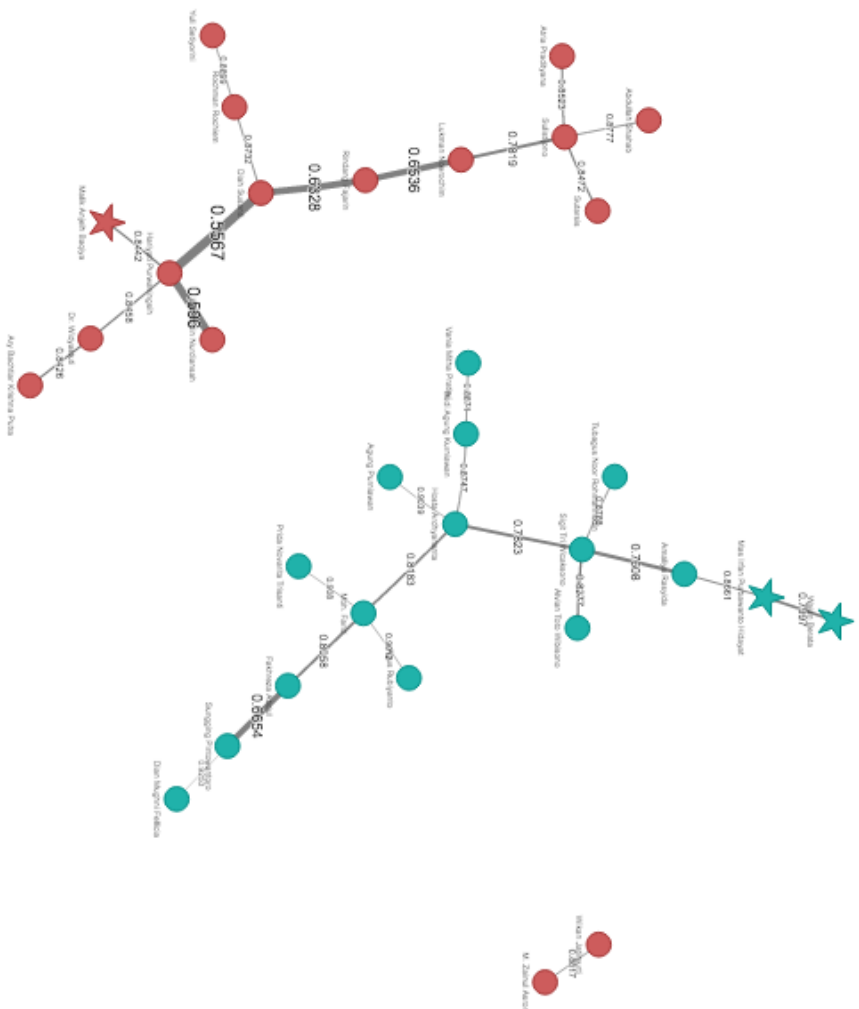
Gambar 8.83 Gambar graf cluster peneliti jurusan Teknik Elektro ITS dengan nilai K sama dengan 16



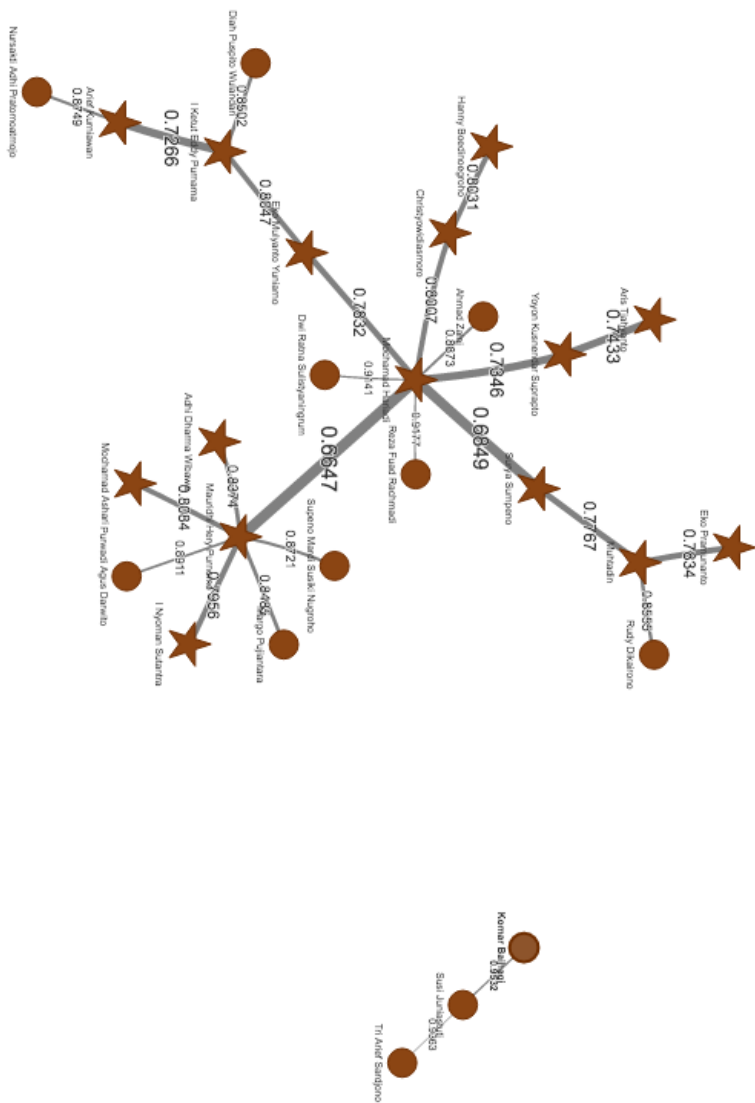
Gambar 8.84 Gambar graf cluster peneliti jurusan Teknik Kimia dengan nilai K sama dengan 16



Gambar 8.85 Gambar graf cluster peneliti jurusan Teknik Fisika dengan nilai K sama dengan 16

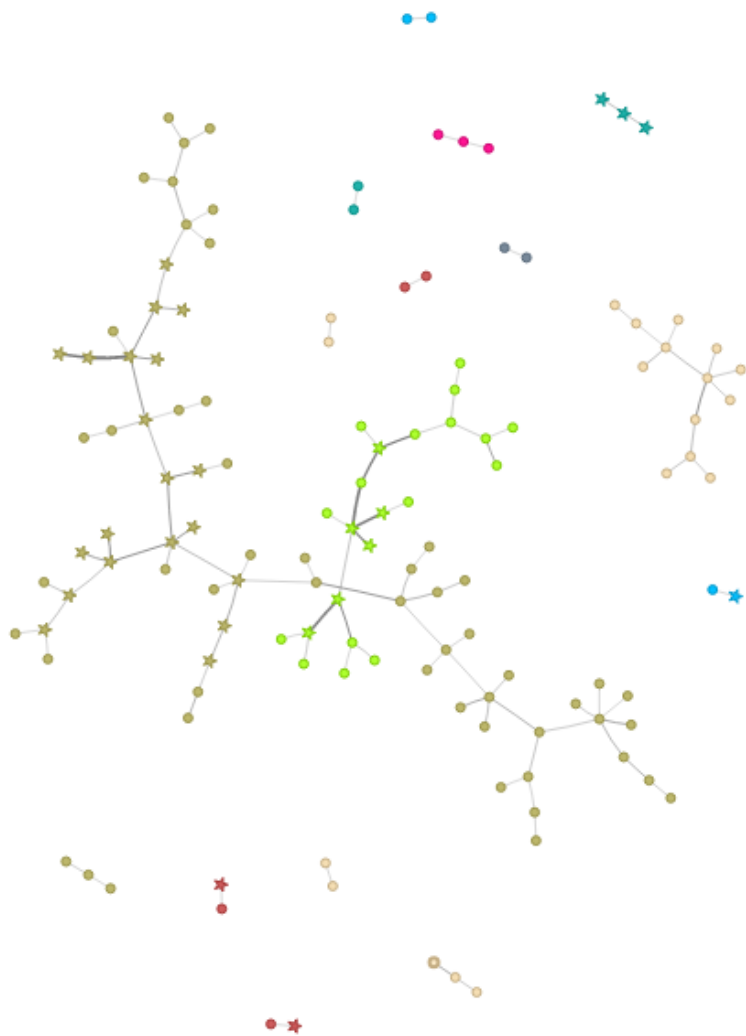


Gambar 8.87 Gambar graf cluster peneliti jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 16

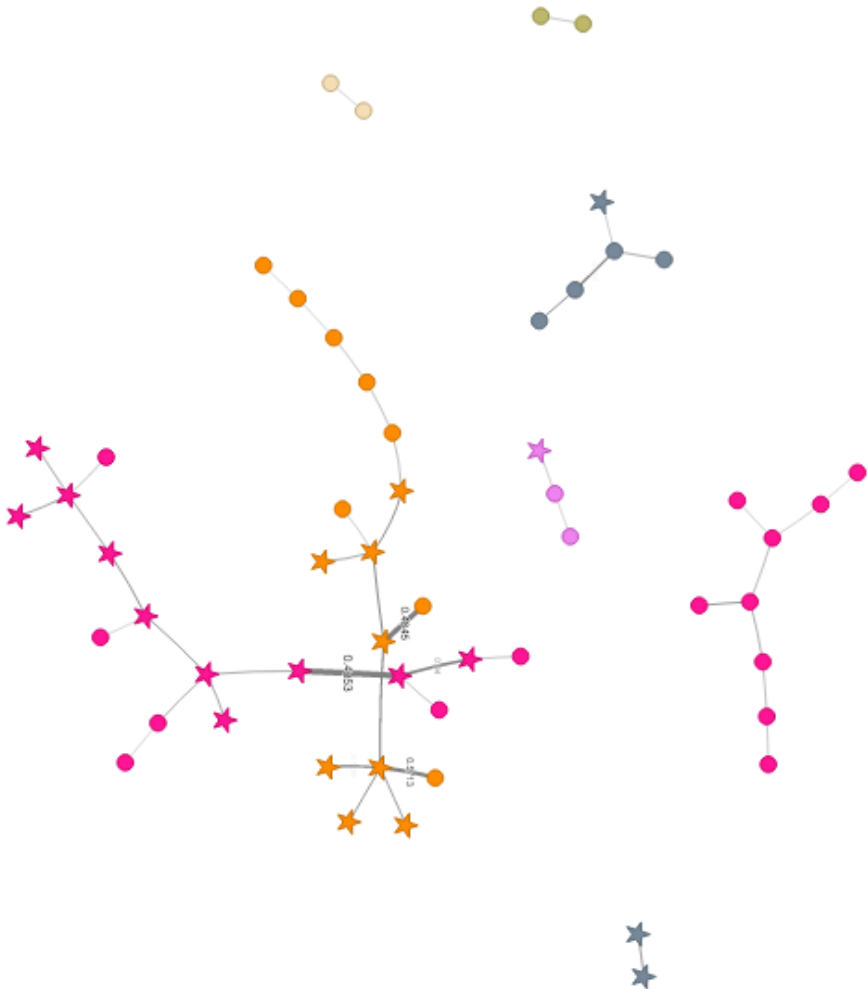


Gambar 8.88 Gambar graf cluster peneliti jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 16

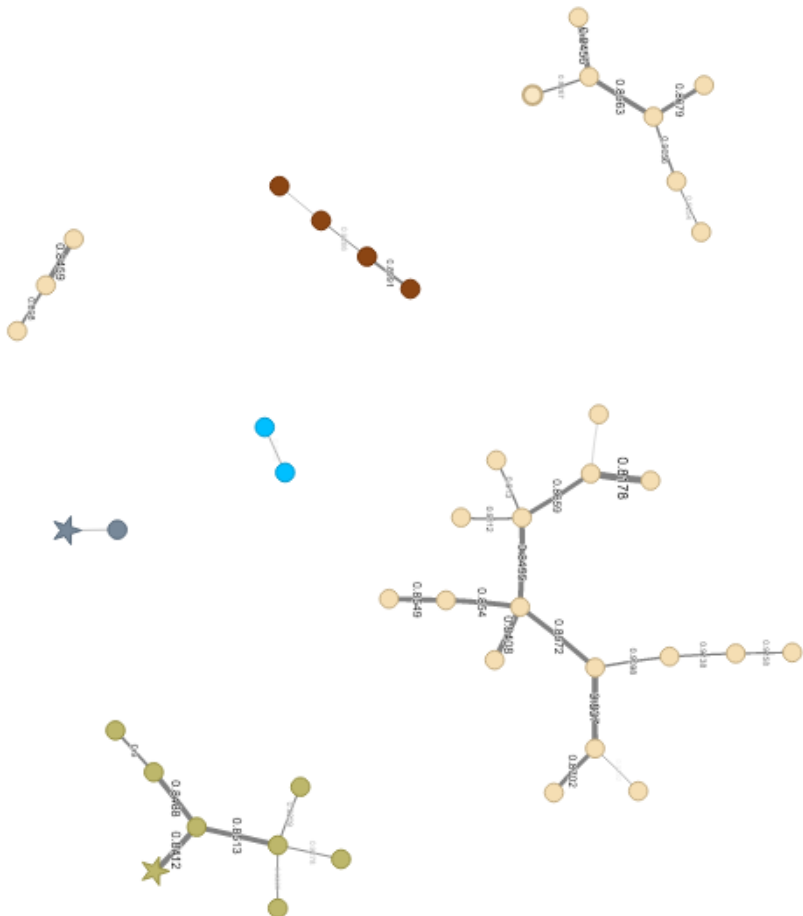
Gambar 8.89 Gambar graf cluster peneliti jurusan Manajemen Bisnis ITS dengan nilai K sama dengan 16



Gambar 8.90 Gambar graf cluster peneliti jurusan Teknik Sipil ITS dengan nilai K sama dengan 16

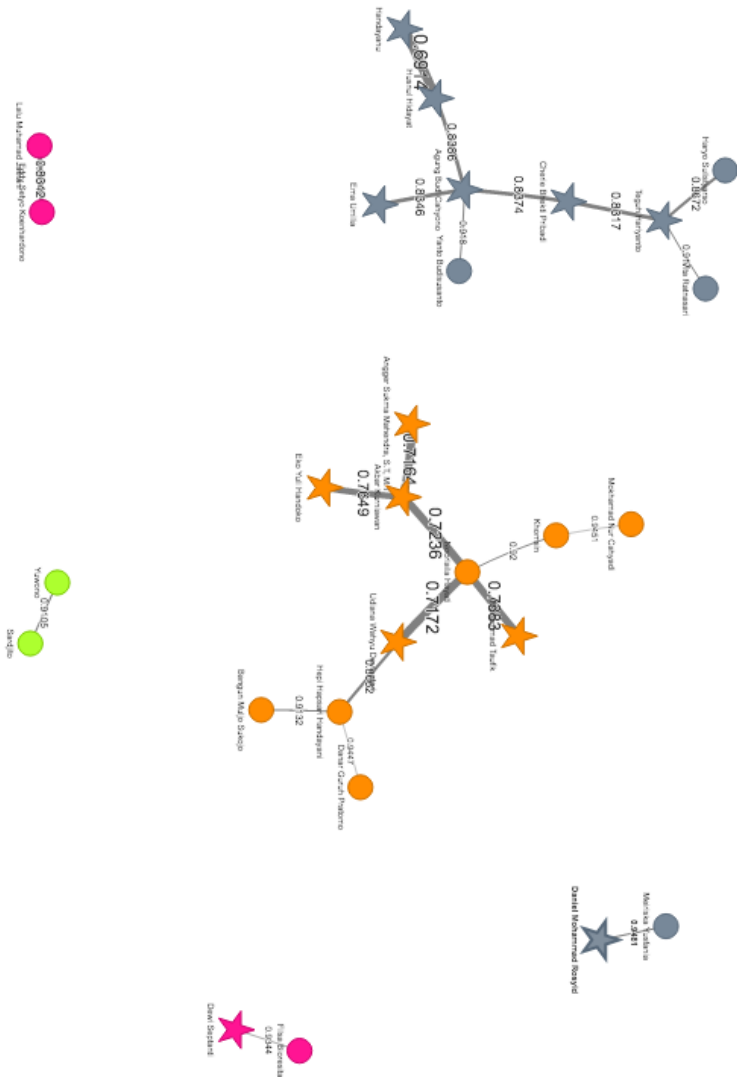


Gambar 8.91 Gambar graf cluster peneliti jurusan Arsitektur ITS dengan nilai K sama dengan 16

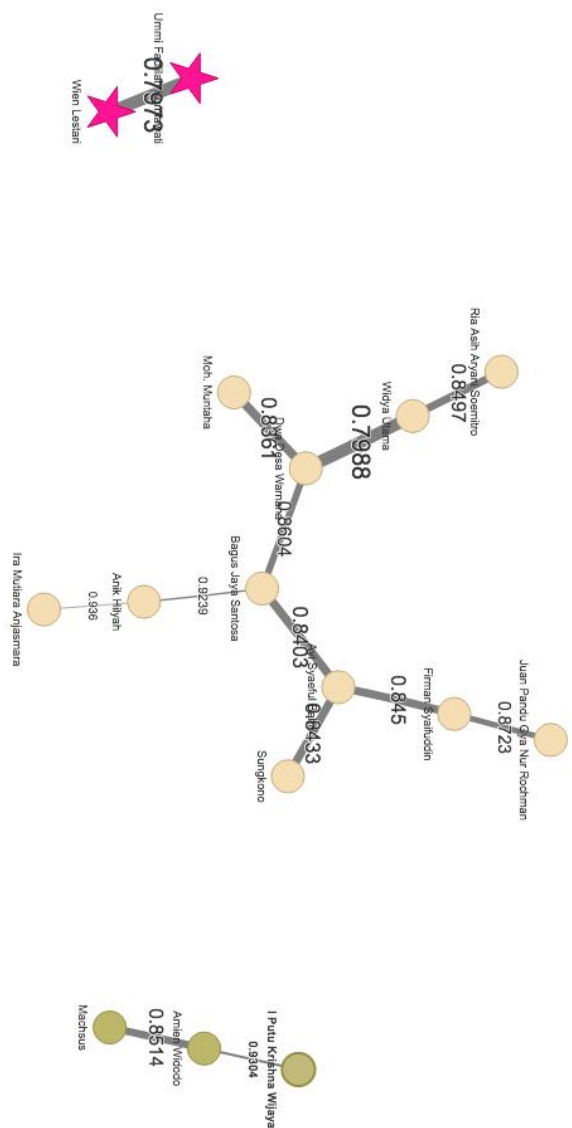


Gambar 8.92 Gambar graf cluster peneliti jurusan Teknik Lingkungan ITS dengan nilai K sama dengan 16

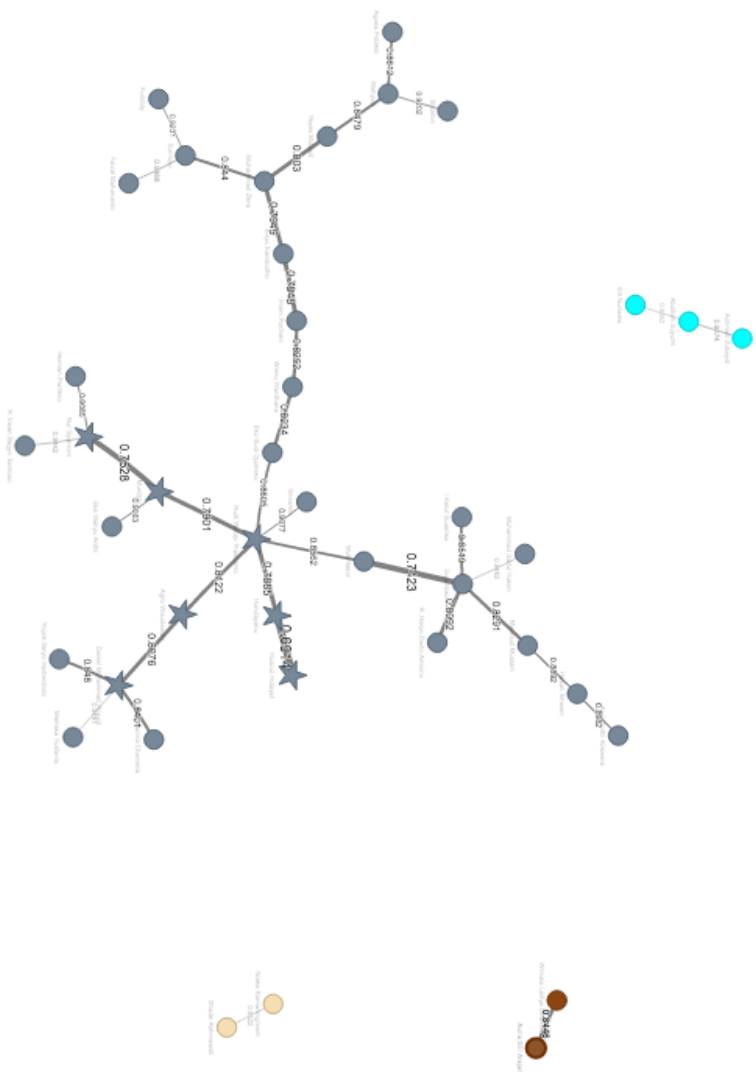
Gambar 8.93 Gambar graf cluster peneliti jurusan Desain Produk ITS dengan nilai K sama dengan 16



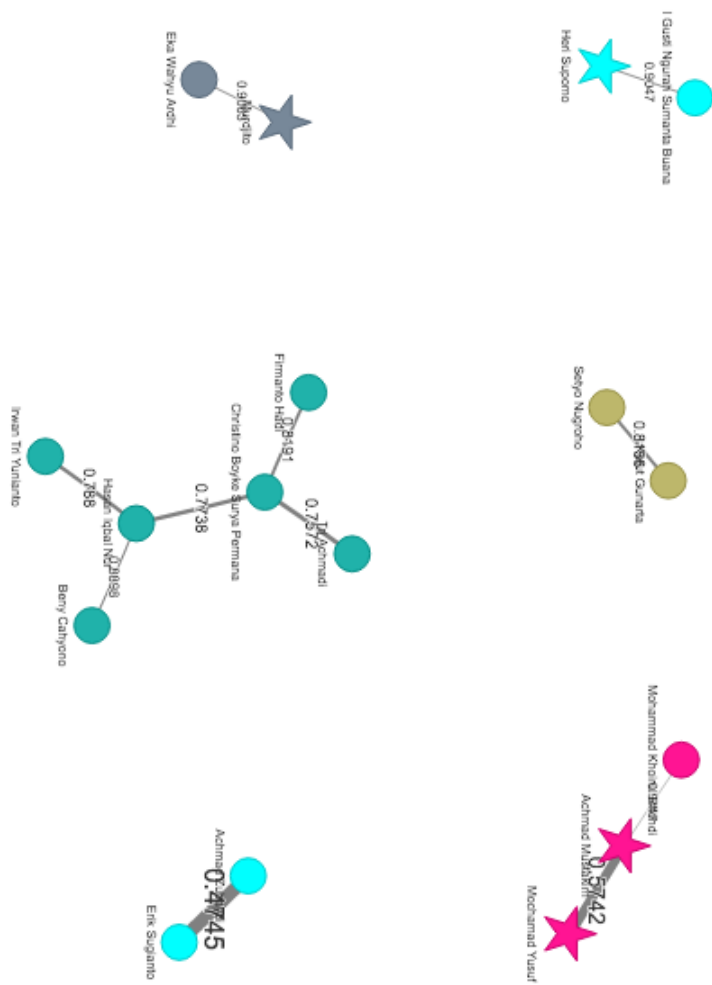
Gambar 8.94 Gambar graf cluster peneliti jurusan Teknik Geomatika ITS dengan nilai K sama dengan 16



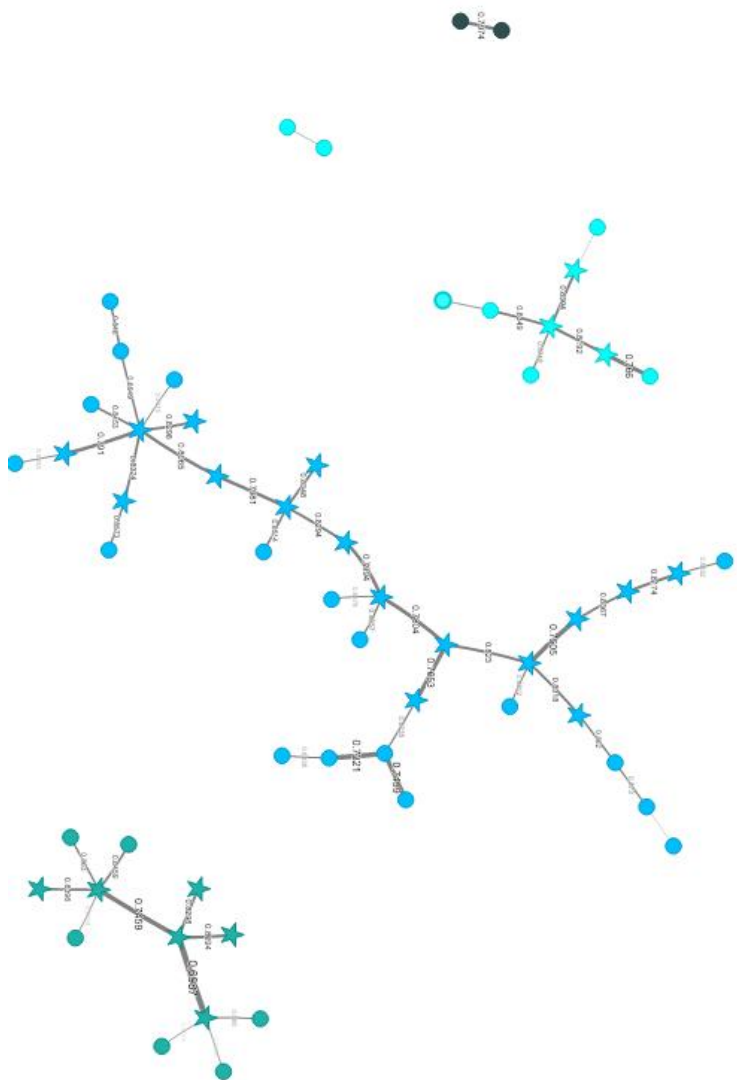
Gambar 8.96 Gambar graf cluster peneliti jurusan Teknik Geofisika ITS dengan nilai K sama dengan 16



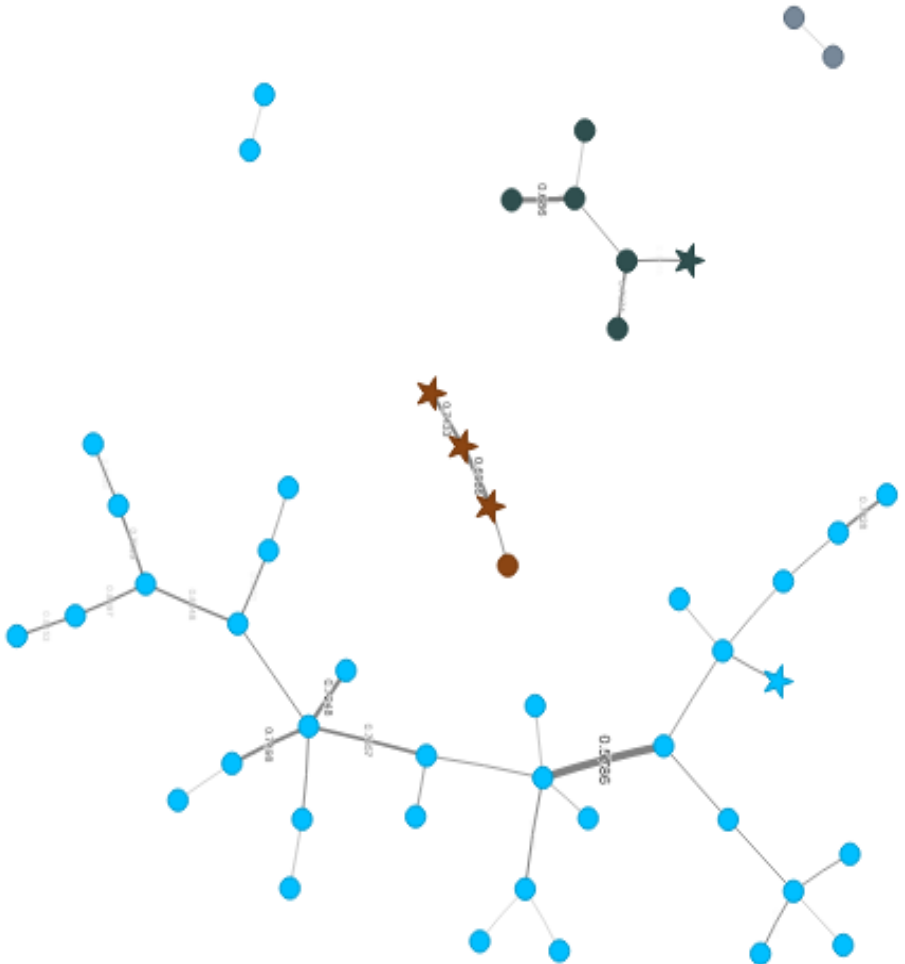
Gambar 8.100 Gambar graf cluster peneliti jurusan Teknik Kelautan ITS dengan nilai K sama dengan 16



Gambar 8.101 Gambar graf cluster peneliti jurusan Transportasi Laut ITS dengan nilai K sama dengan 16

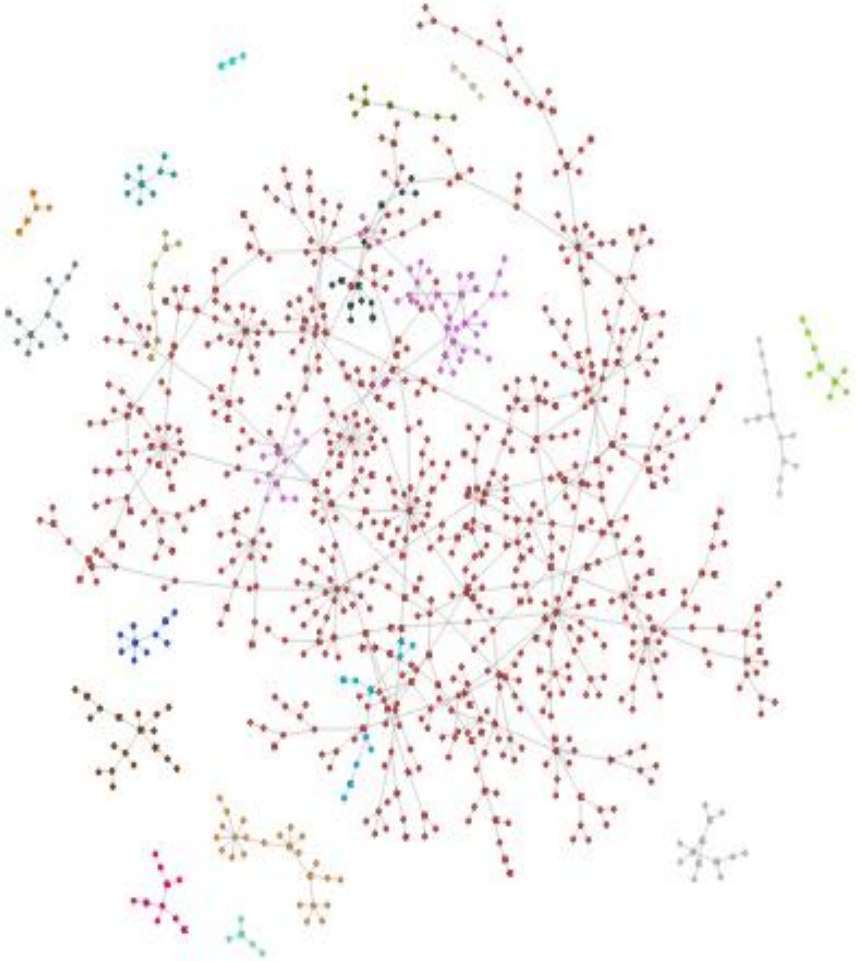


Gambar 8.102 Gambar graf cluster peneliti jurusan Teknik Informatika ITS dengan nilai K sama dengan 16

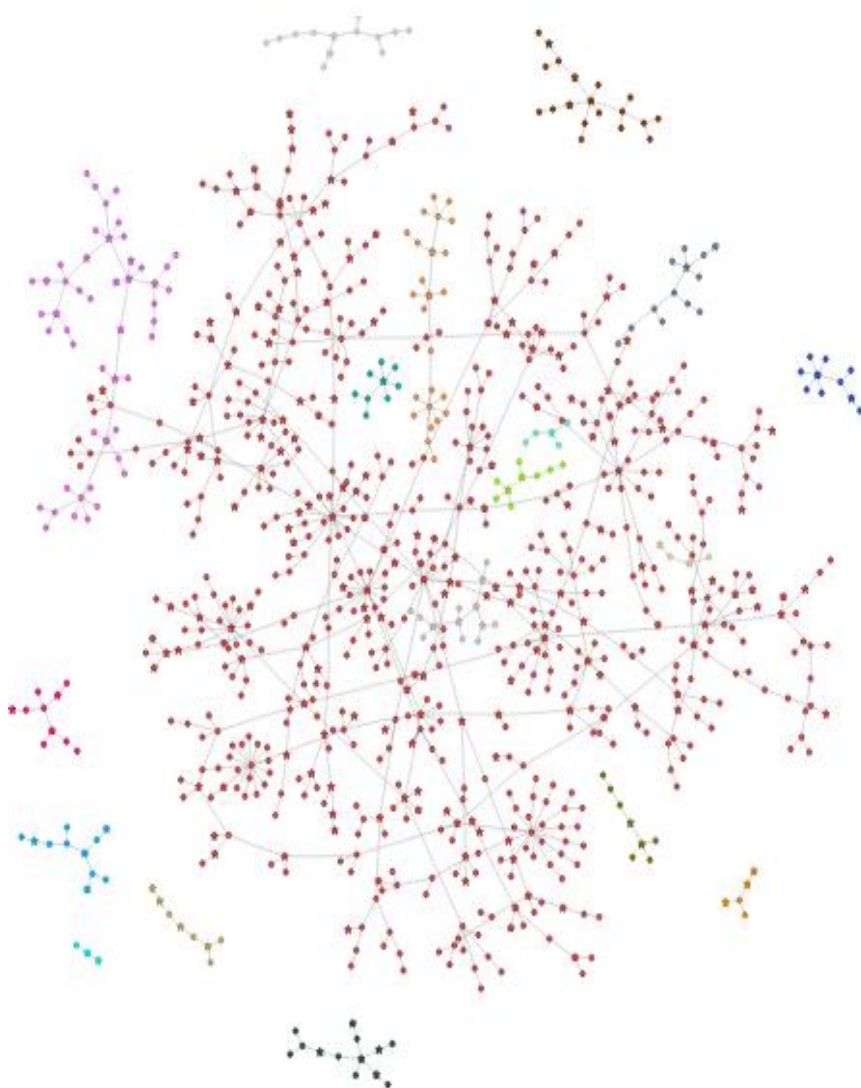


Gambar 8.103 Gambar graf cluster peneliti jurusan Sistem Informasi ITS dengan nilai K sama dengan 16

8.1.4 Gambar Graf Cluster Kerjasama Peneliti Ekspansi Sinonim Kata dengan Nilai K sama dengan 10



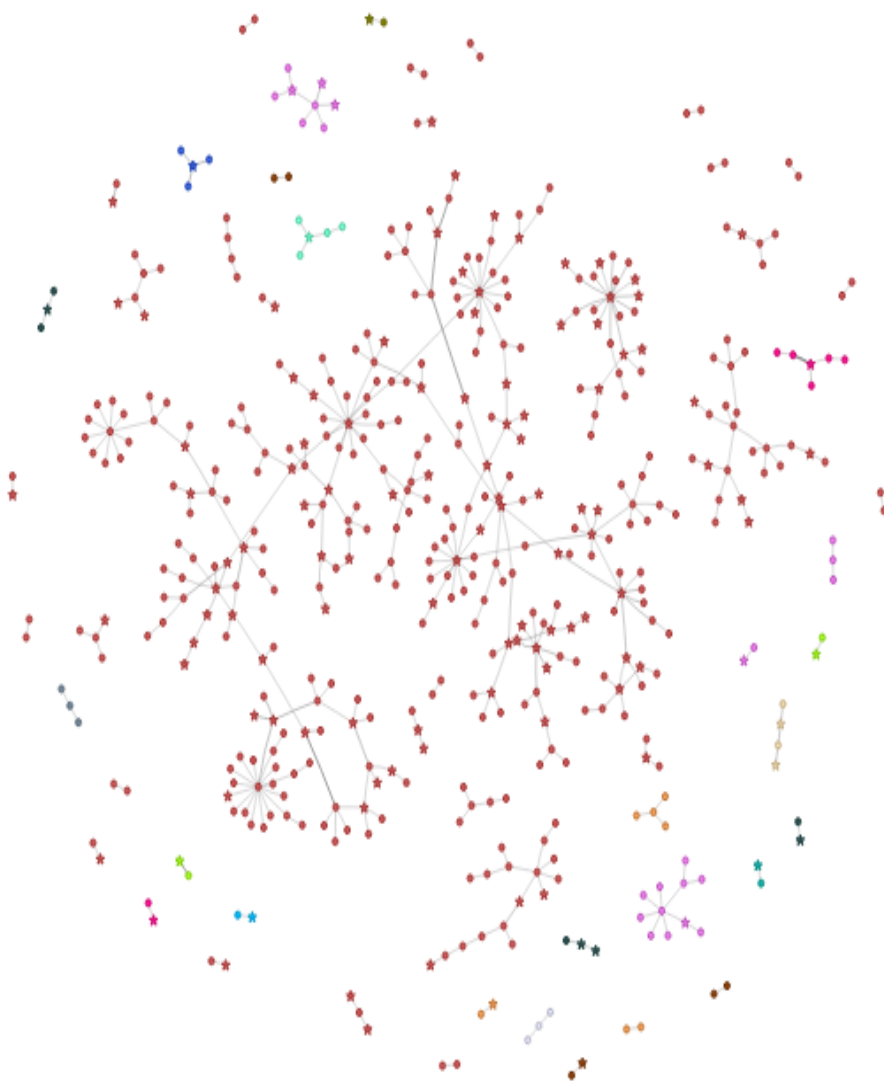
Gambar 8.104 Gambar graf cluster peneliti ITS ekspansi sinonim kata dengan K sama dengan 10



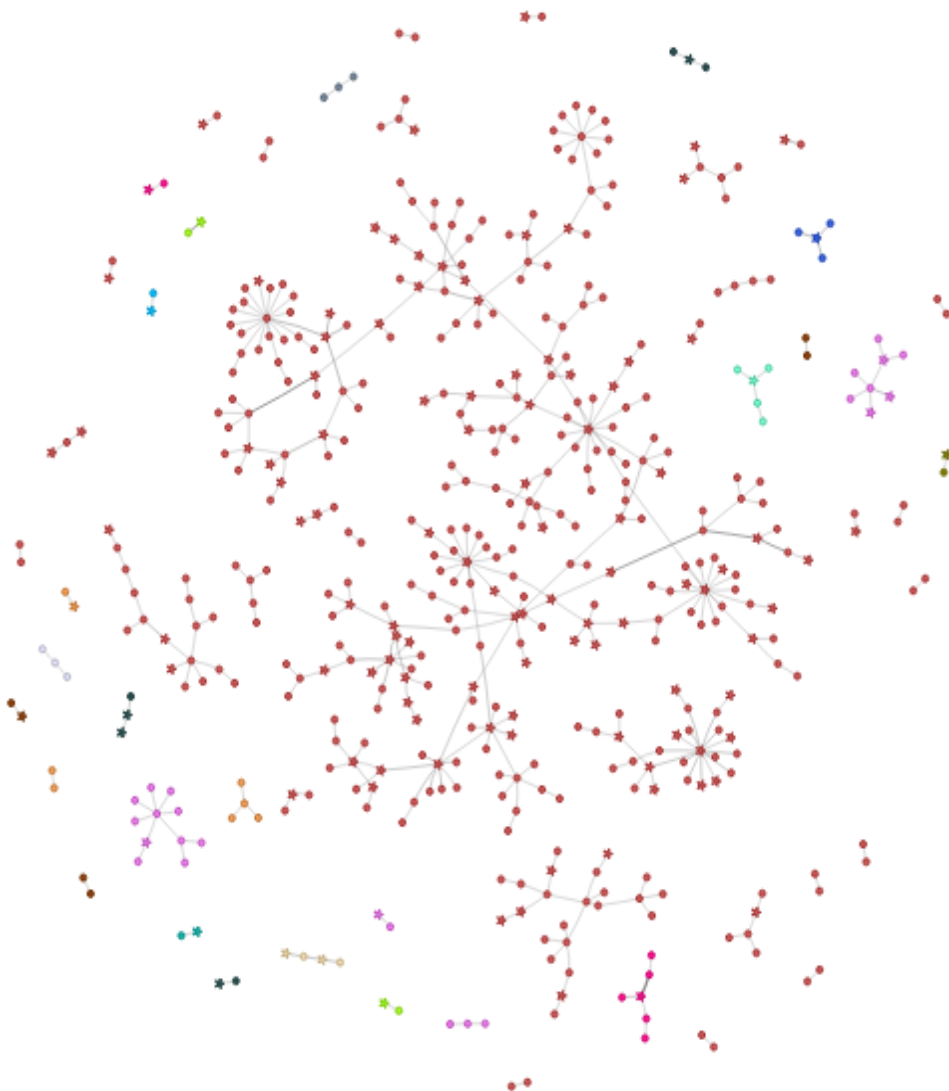
Gambar 8.105 Gambar graf cluster peneliti ekspansi sinonim kata FMIPA ITS dengan K sama dengan 10



Gambar 8.106 Gambar graf cluster peneliti ekspansi sinonim kata FTI ITS dengan K sama dengan 10

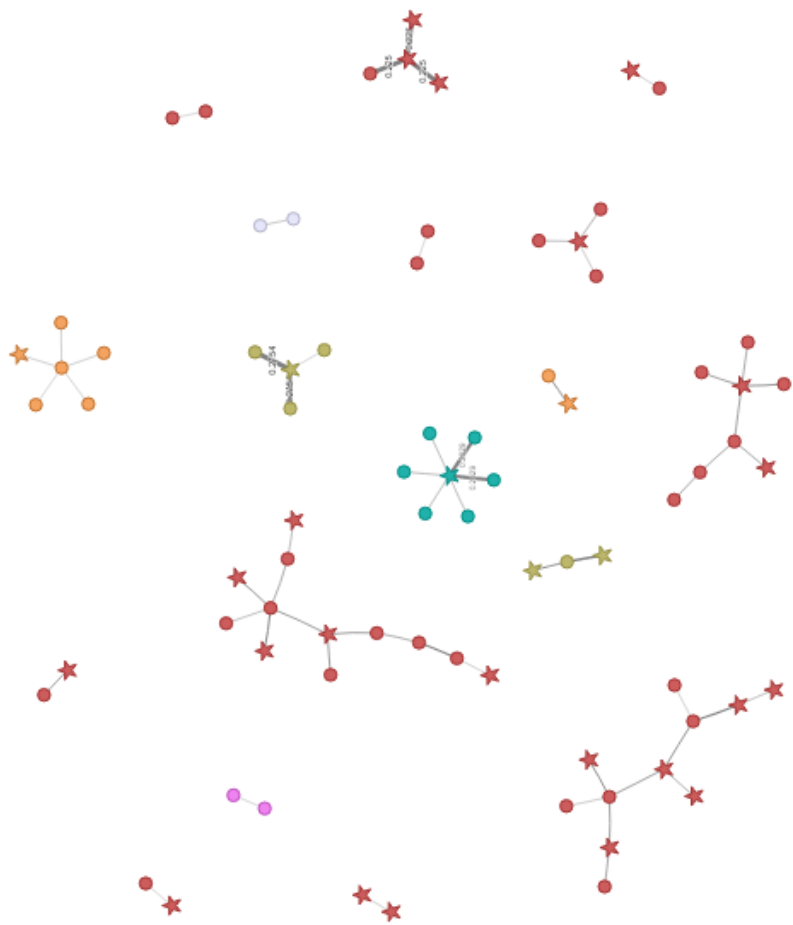


Gambar 8.107 Gambar graf peneliti ekspansi sinonim kata FTSP ITS dengan K sama dengan 10

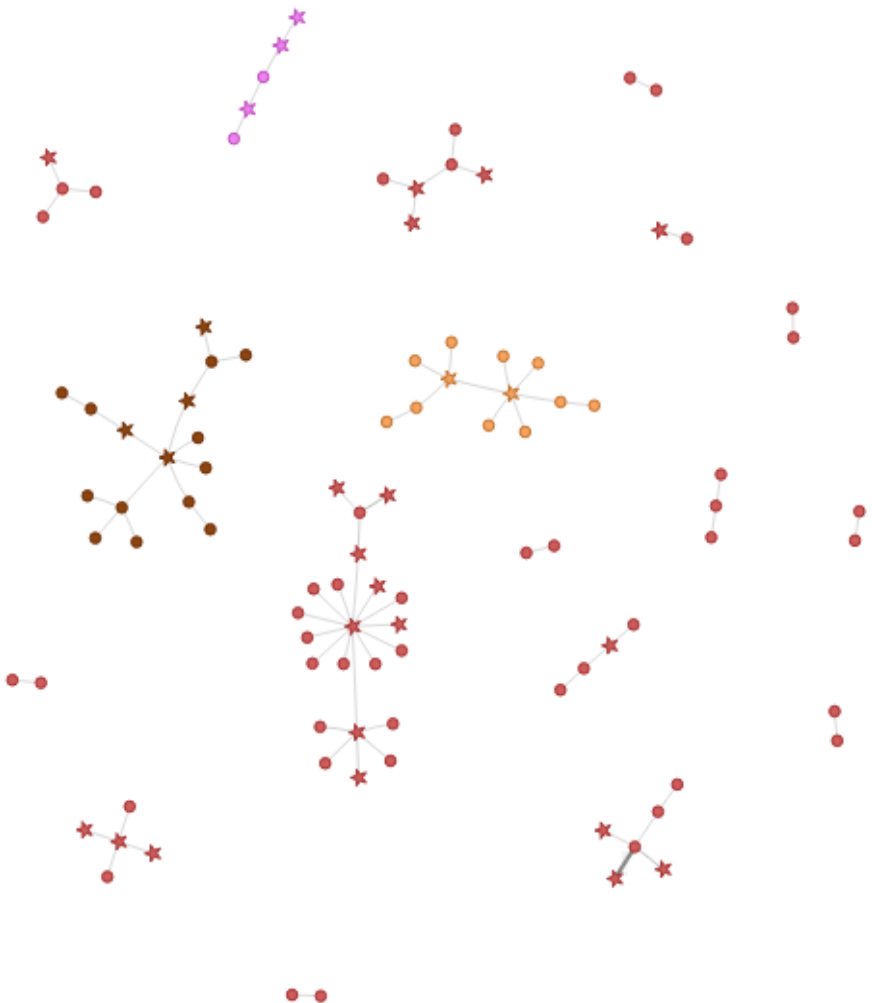


Gambar 8.108 Gambar graf cluster peneliti ekspansi sinonim kata FTK ITS dengan nilai K sama dengan 10

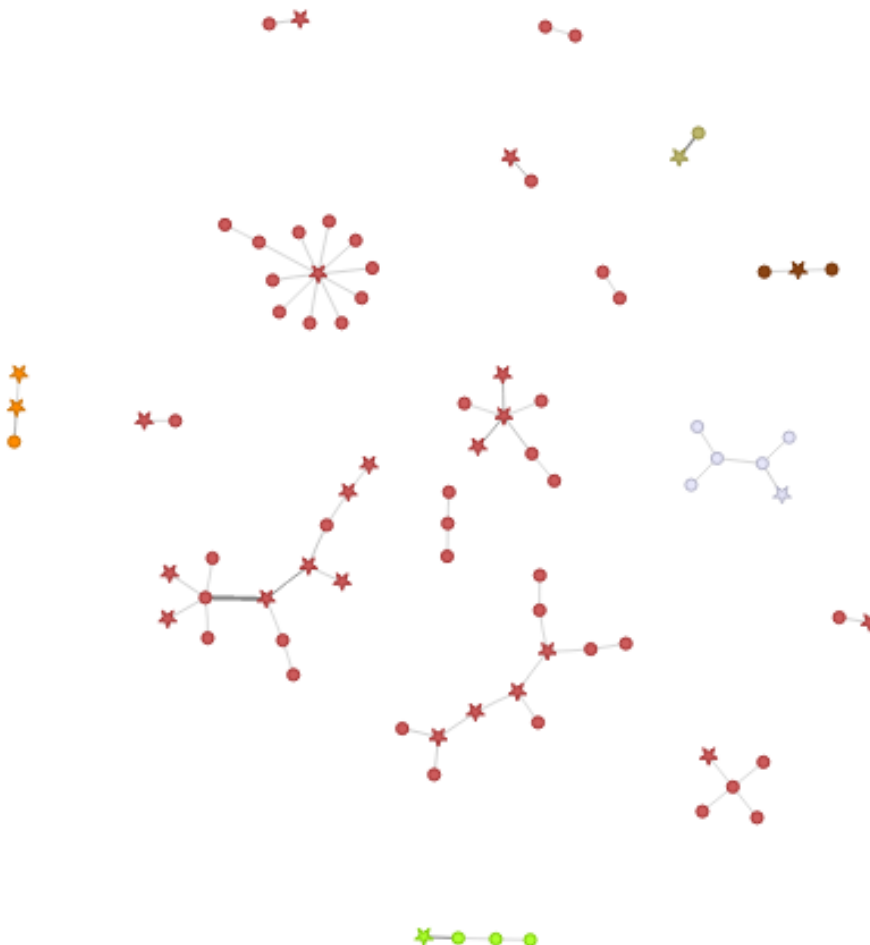
Gambar 8.109 Gambar graf cluster peneliti ekspansi sinonim kata FTIF ITS dengan nilai K sama dengan 10



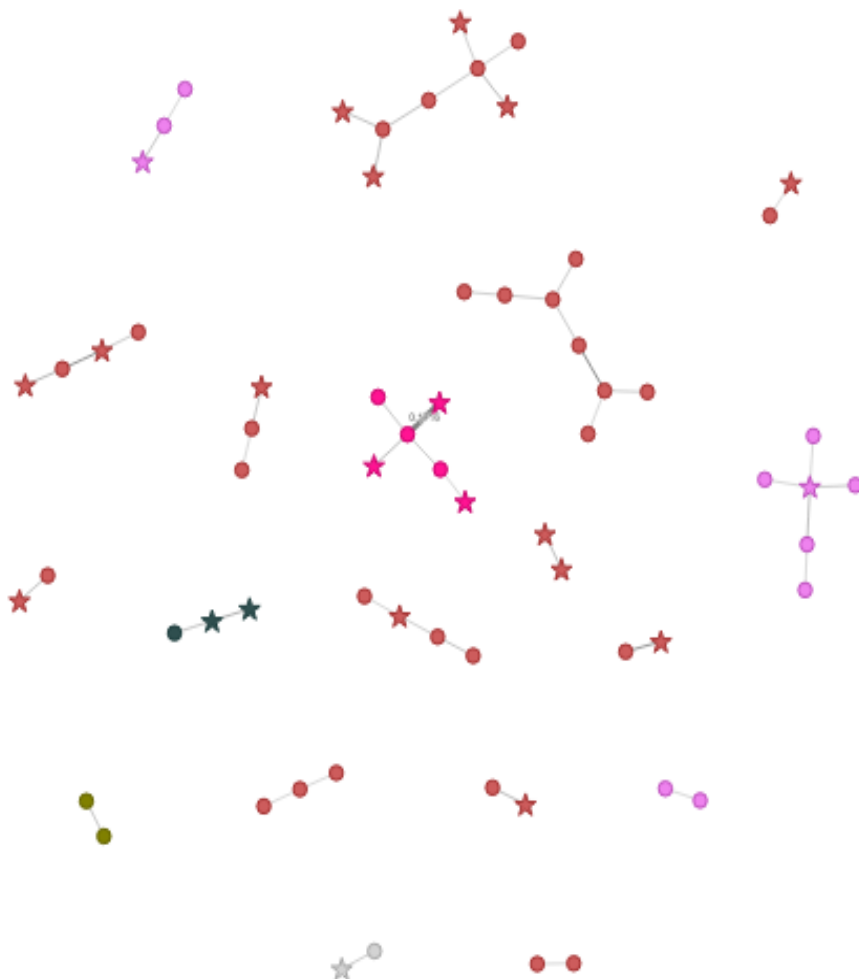
Gambar 8.110 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Fisika ITS dengan nilai K sama dengan 10



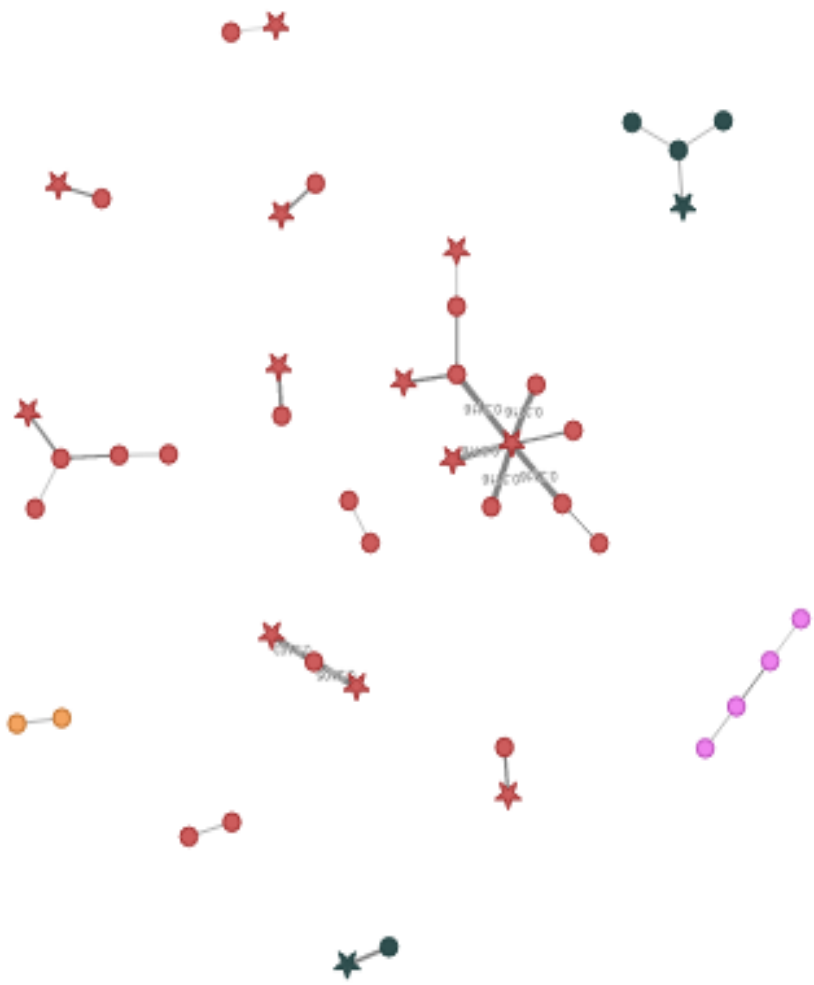
Gambar 8.111 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Matematika ITS dengan nilai K sama dengan 10



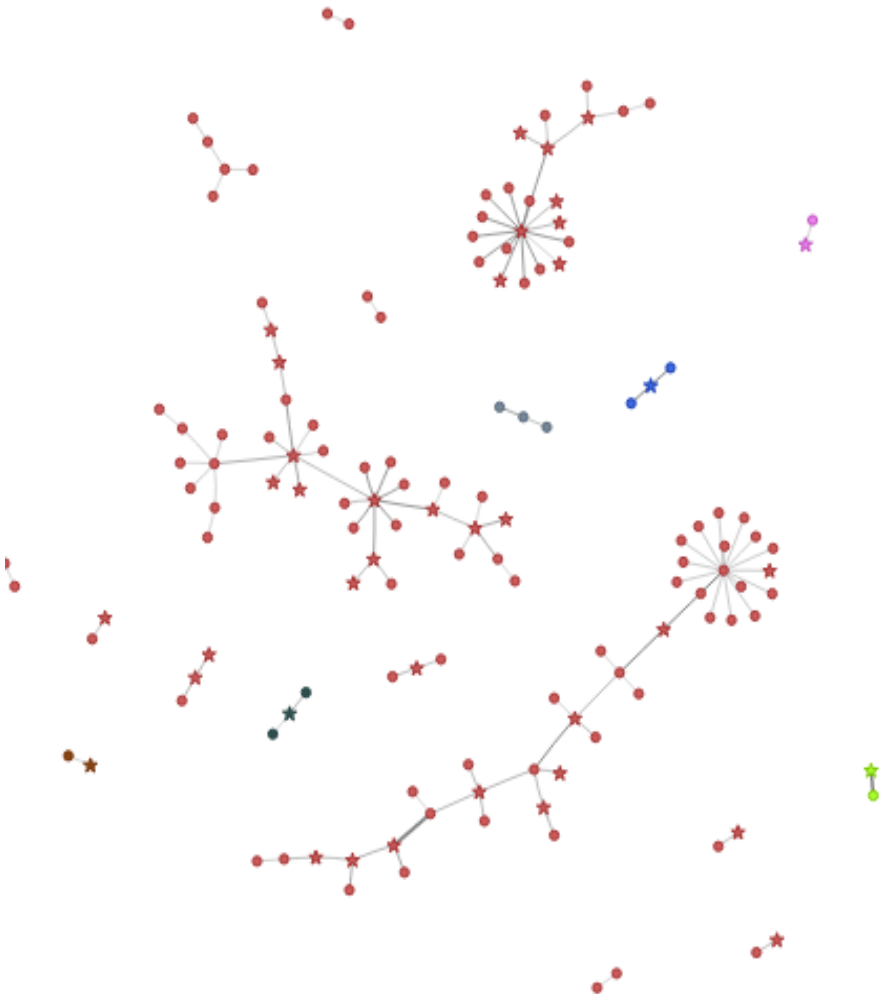
Gambar 8.112 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Statistika ITS dengan nilai K sama dengan 10



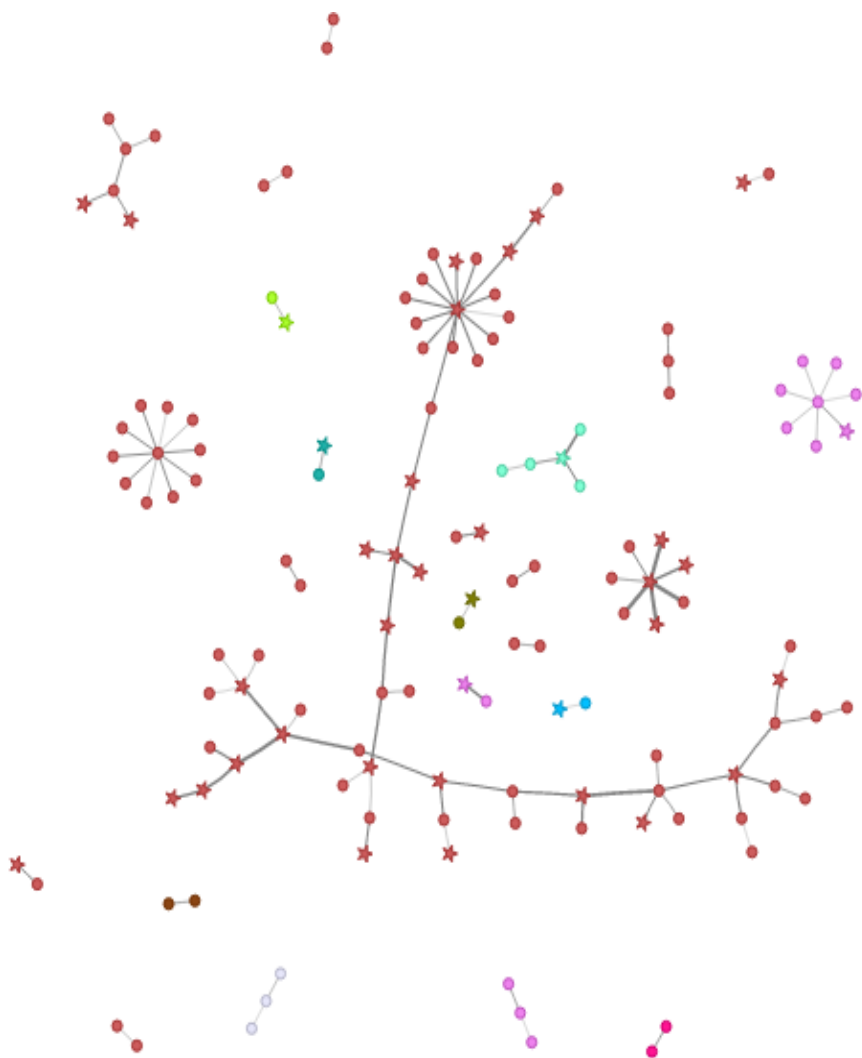
Gambar 8.113 Gambar graf clsuter peneliti ekspansi sinonim kata jurusan Kimia ITS dengan nilai K sama dengan 10



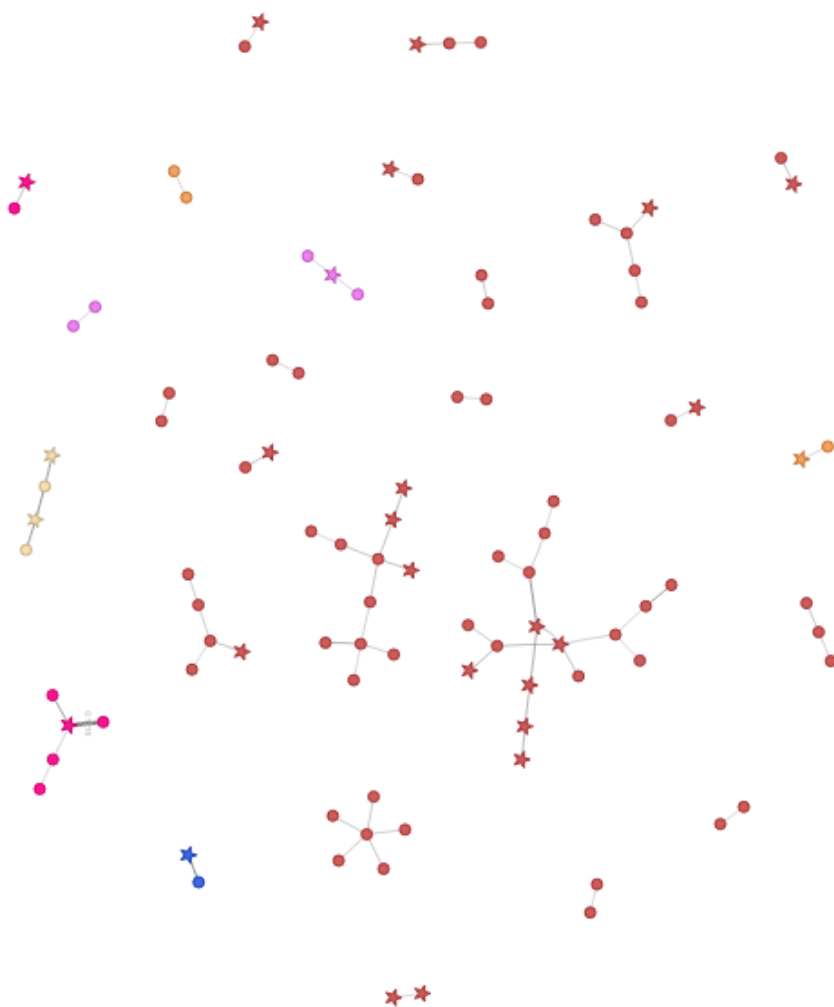
Gambar 8.114 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Biologi ITS dengan nilai K sama dengan 10



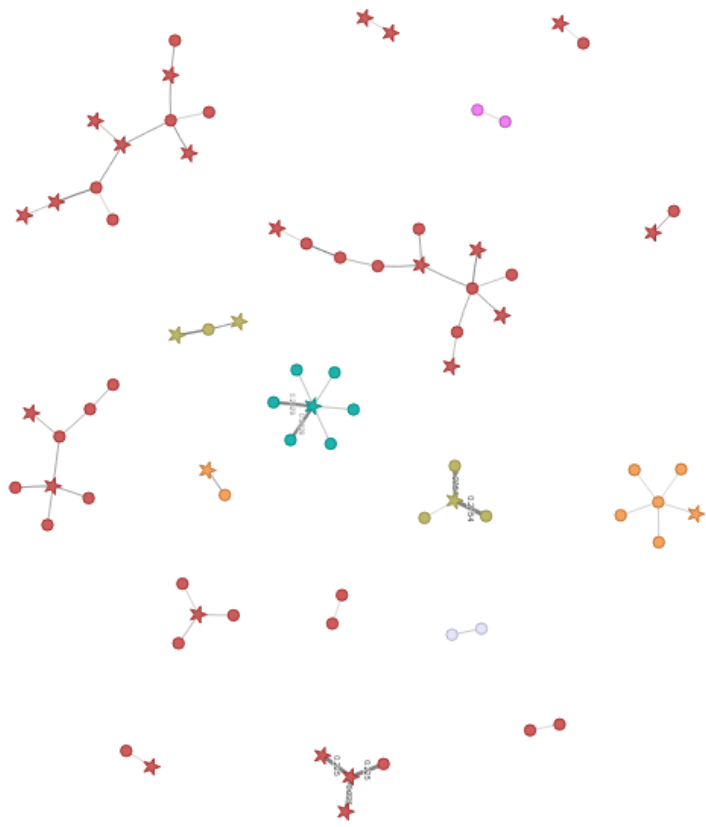
Gambar 8.115 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Mesin ITS dengan nilai K sama dengan 10



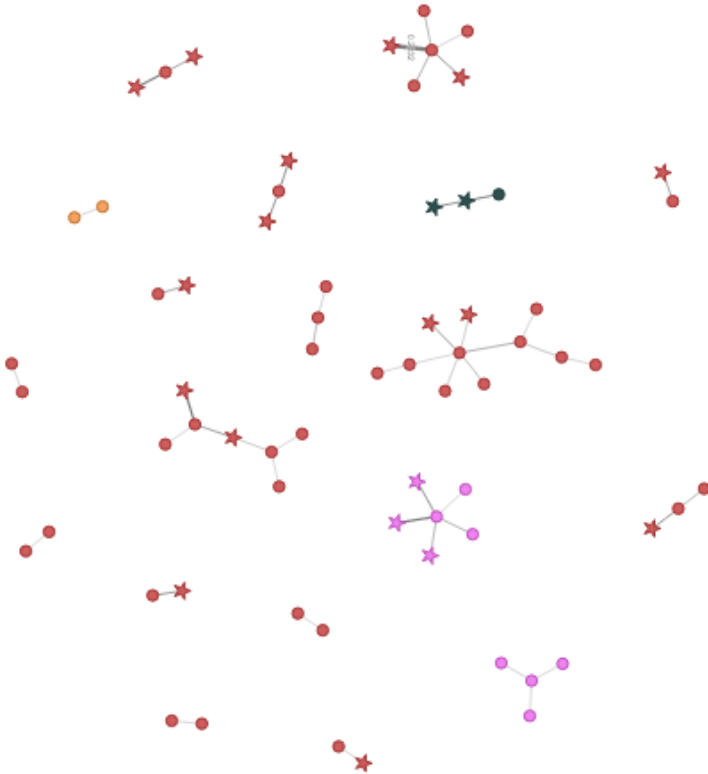
Gambar 8.116 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Elektro ITS dengan nilai K sama dengan 10



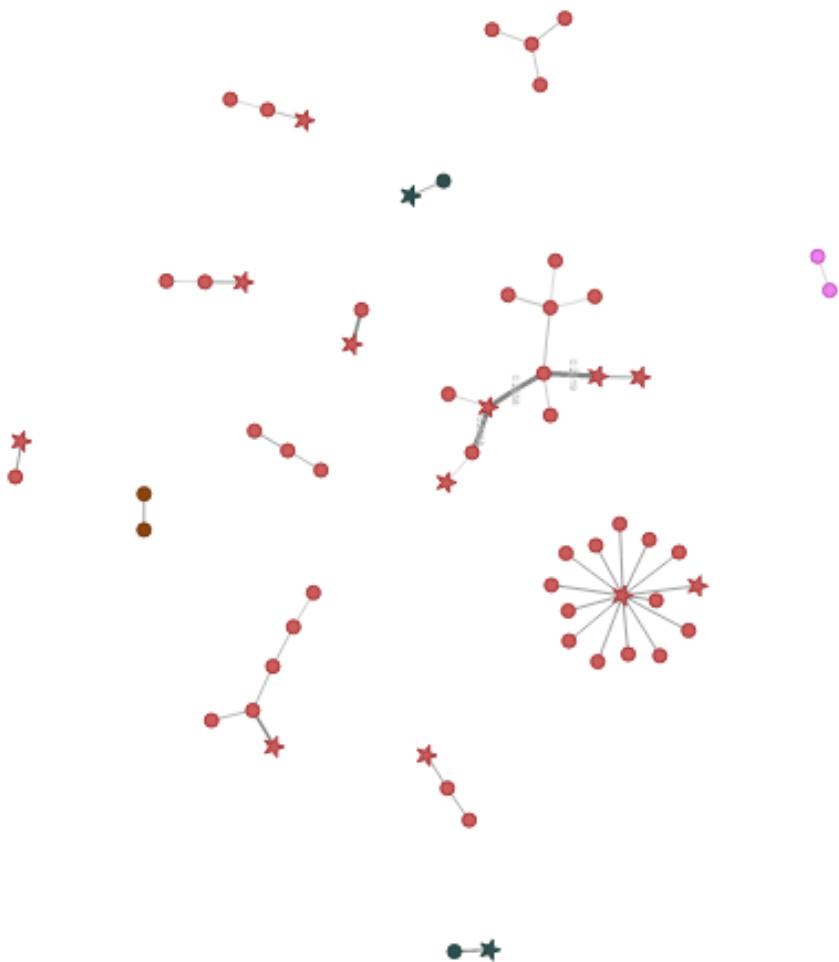
Gambar 8.117 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Kimia dengan nilai K sama dengan 10



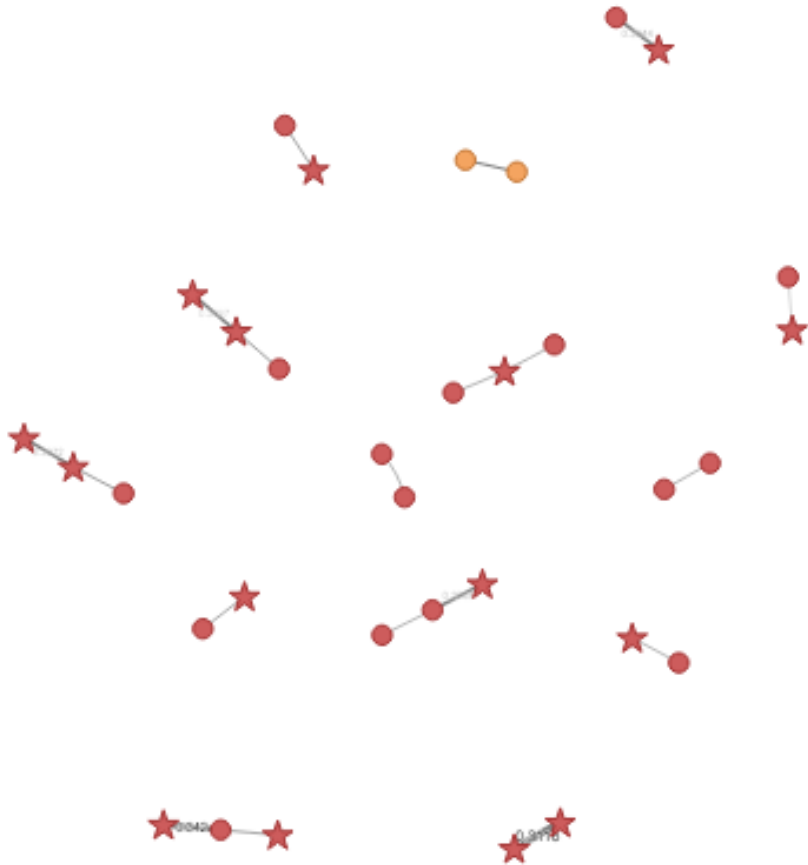
Gambar 8.118 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Fisika dengan nilai K sama dengan 10



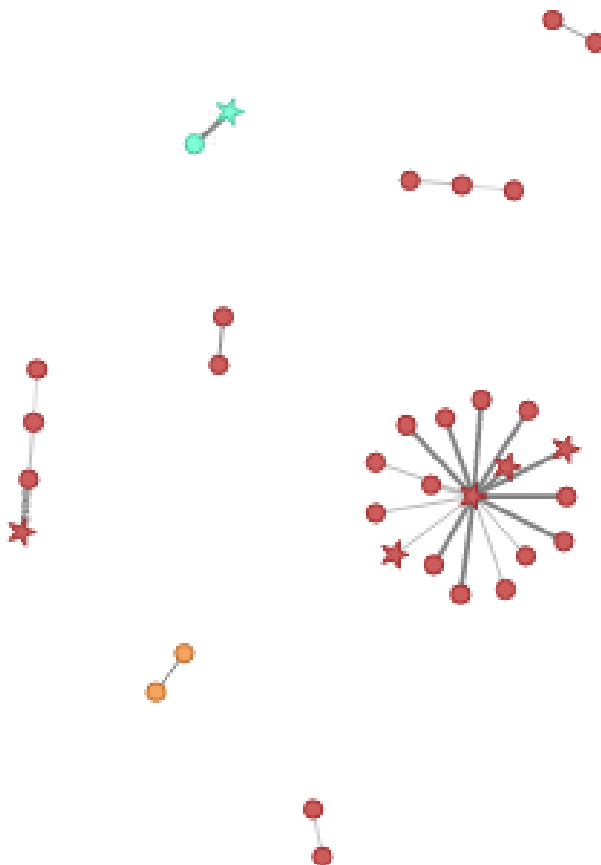
Gambar 8.119 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Industri ITS dengan nilai K sama dengan 10



Gambar 8.120 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Material dan Metalurgi ITS dengan nilai K sama dengan 10



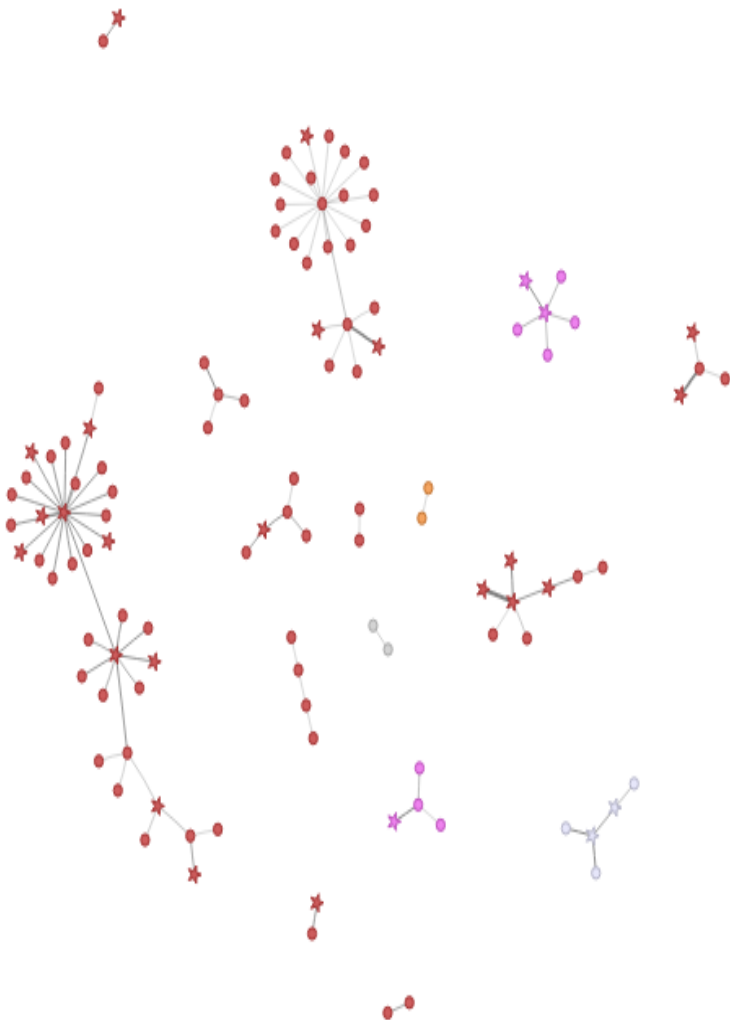
Gambar 8.121 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Multimedia dan Jaringan ITS dengan nilai K sama dengan 10



Gambar 8.122 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Manajemen Bisnis ITS dengan nilai K sama dengan

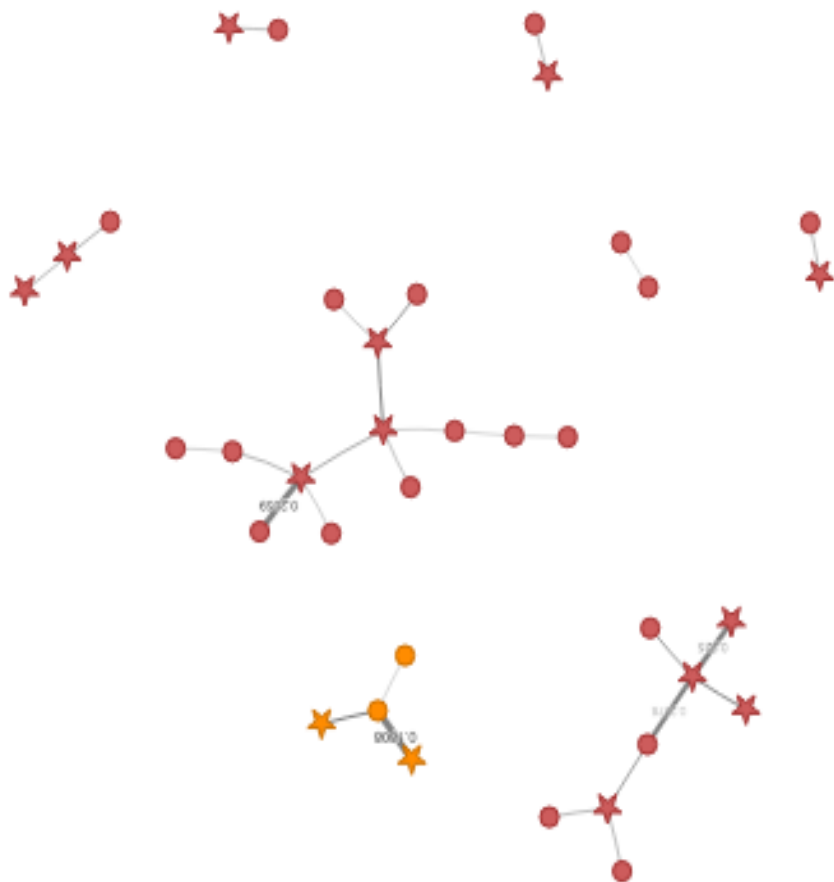


Gambar 8.123 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Sipil ITS dengan nilai K sama dengan 10

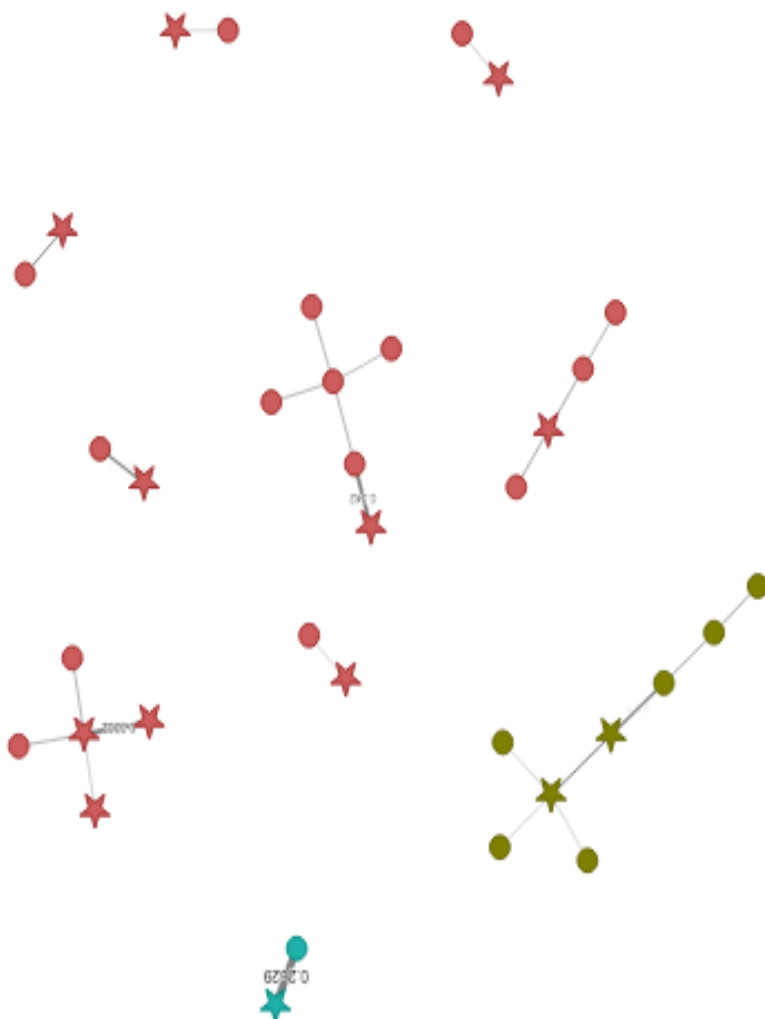


Gambar 8.124 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Arsitektur ITS dengan nilai K sama dengan 10

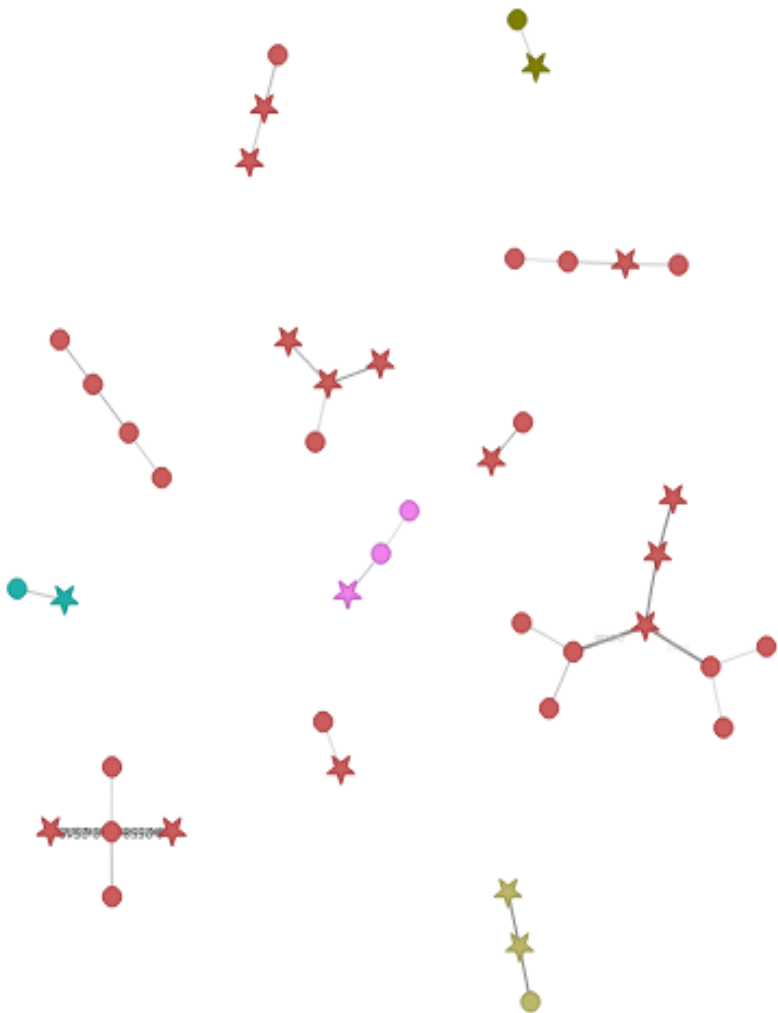
10



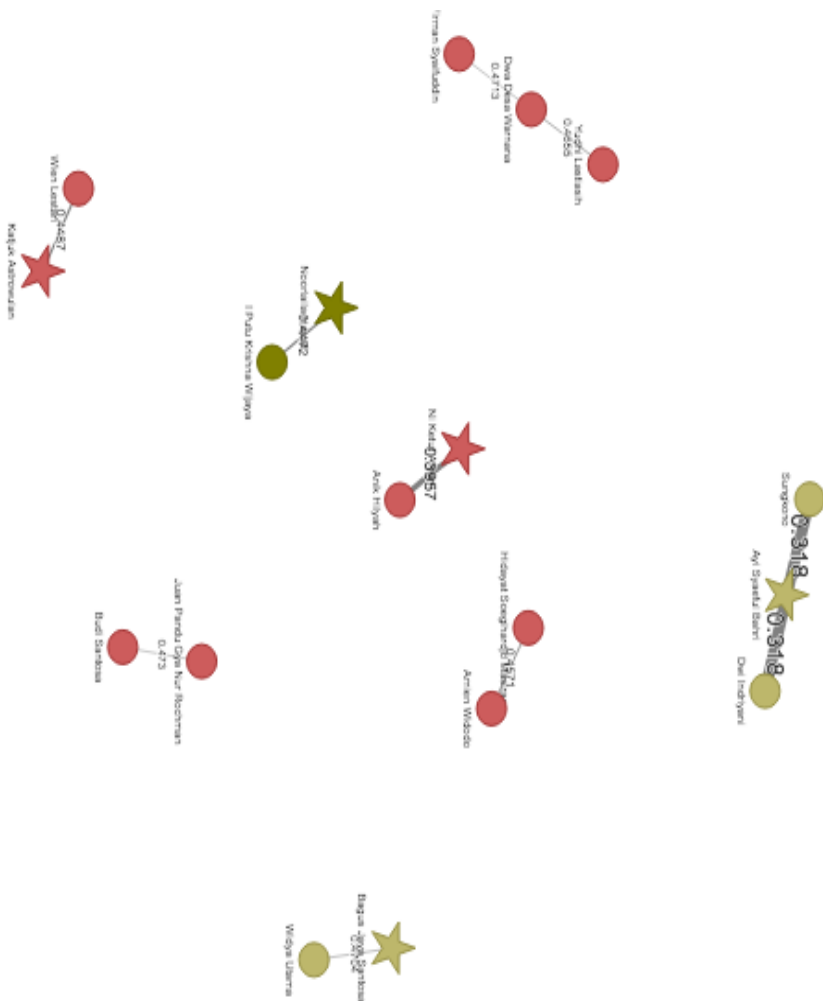
Gambar 8.126 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Desain Produk ITS dengan nilai K sama dengan 10



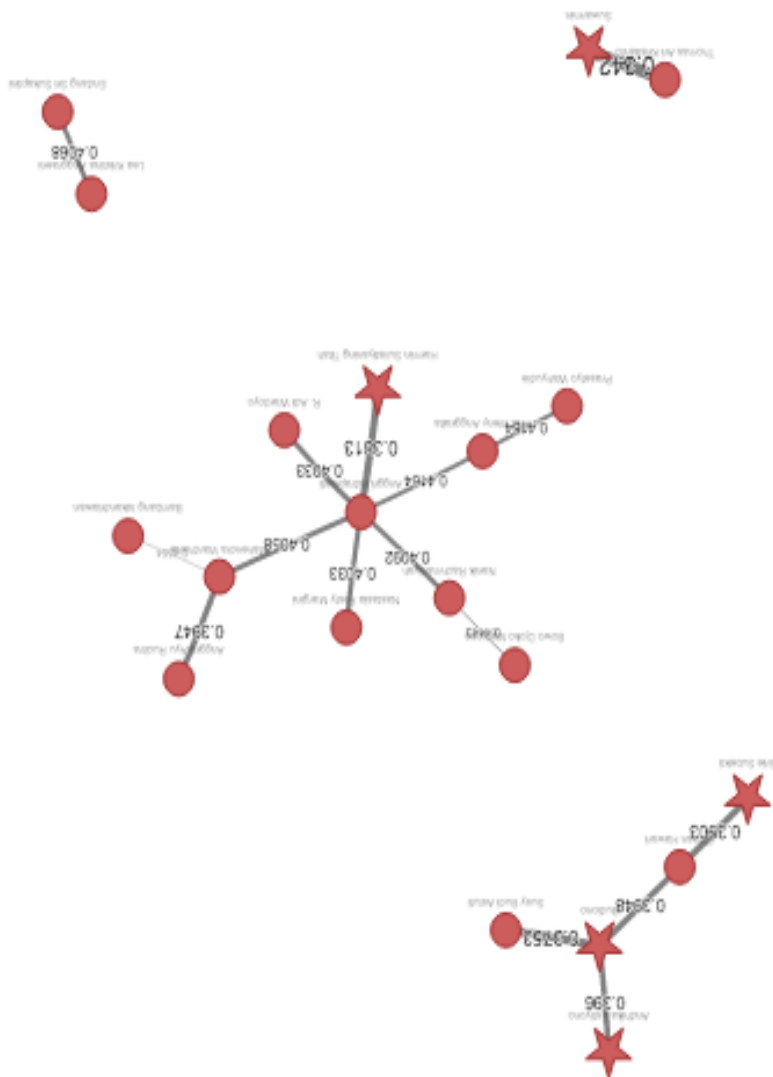
Gambar 8.127 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Geomatika ITS dengan nilai K sama dengan



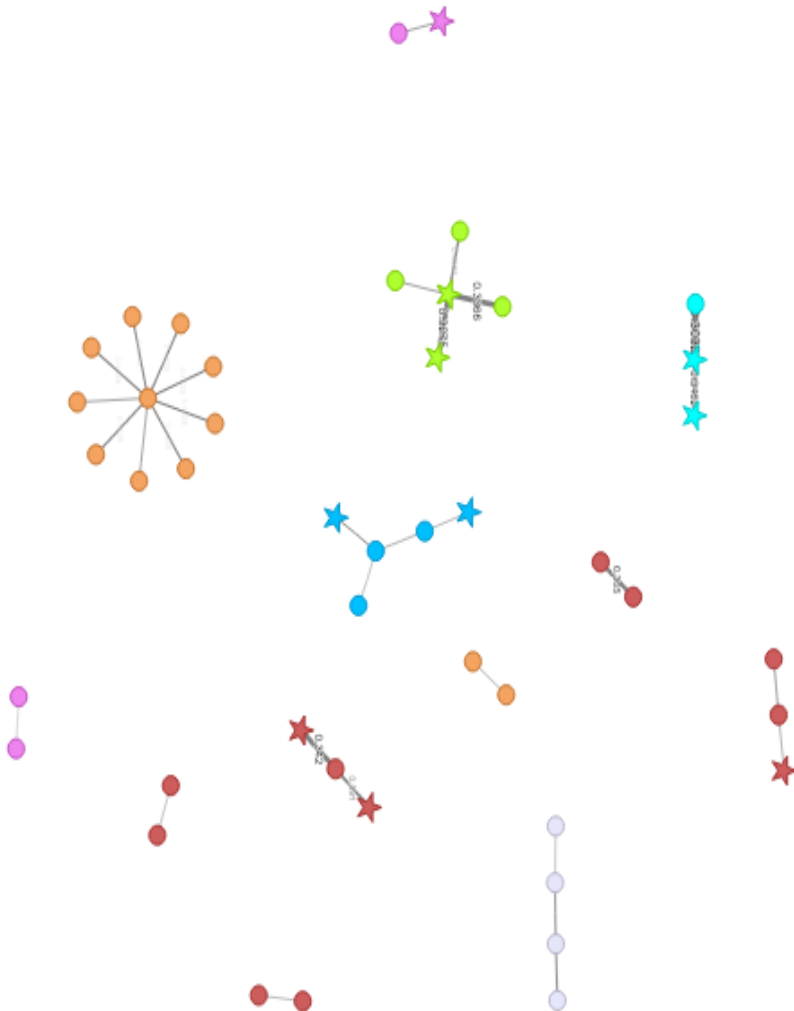
Gambar 8.128 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Perencanaan Wilayah dan Kota ITS dengan nilai K sama dengan 10



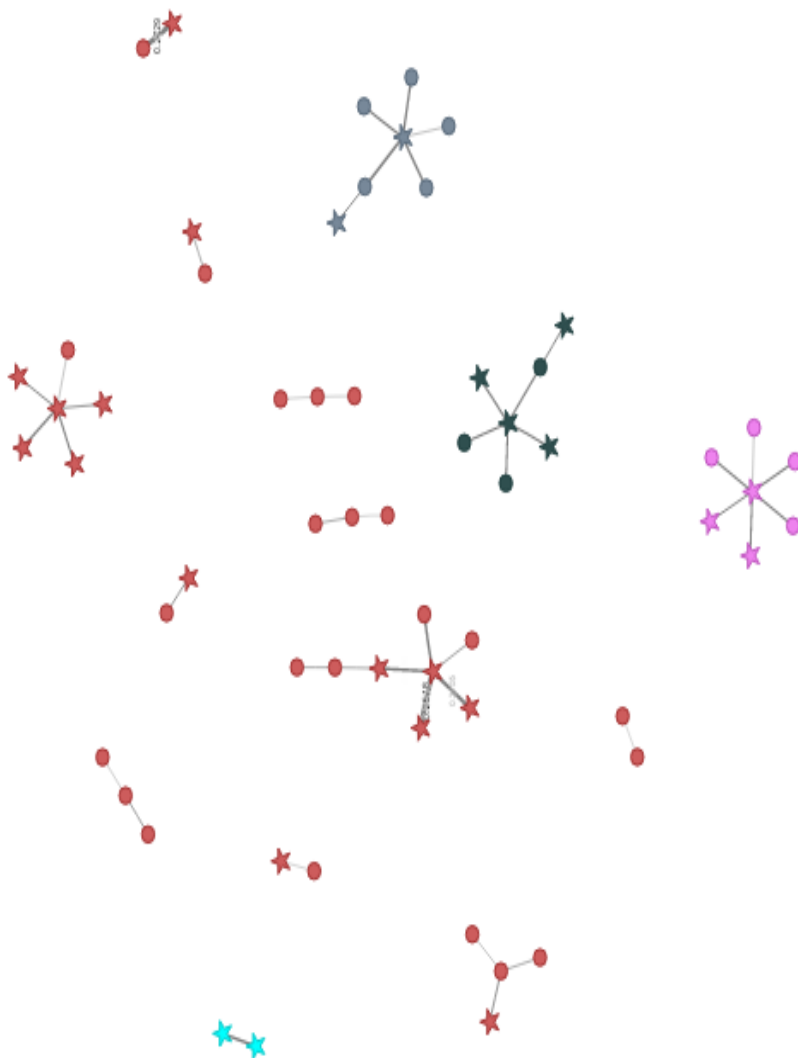
Gambar 8.129 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Geofisika ITS dengan nilai K sama dengan



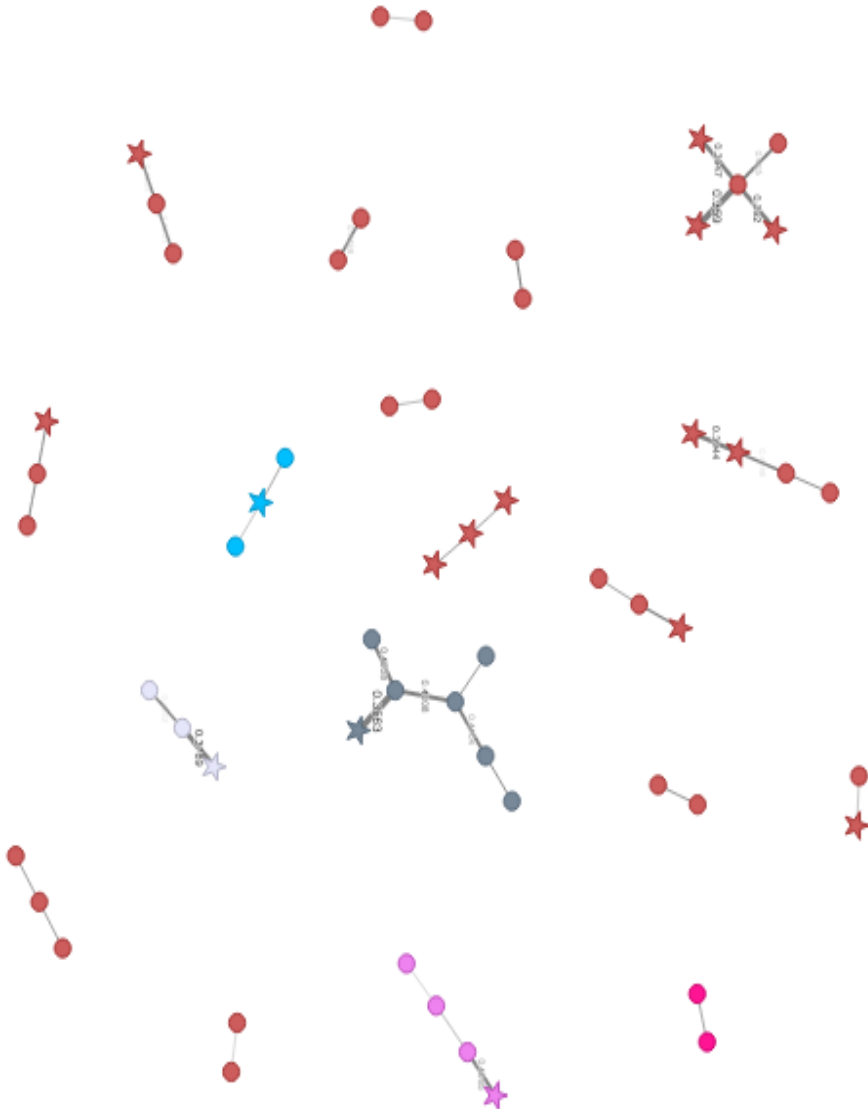
Gambar 8.130 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Desain Interior ITS dengan nilai K sama dengan 10



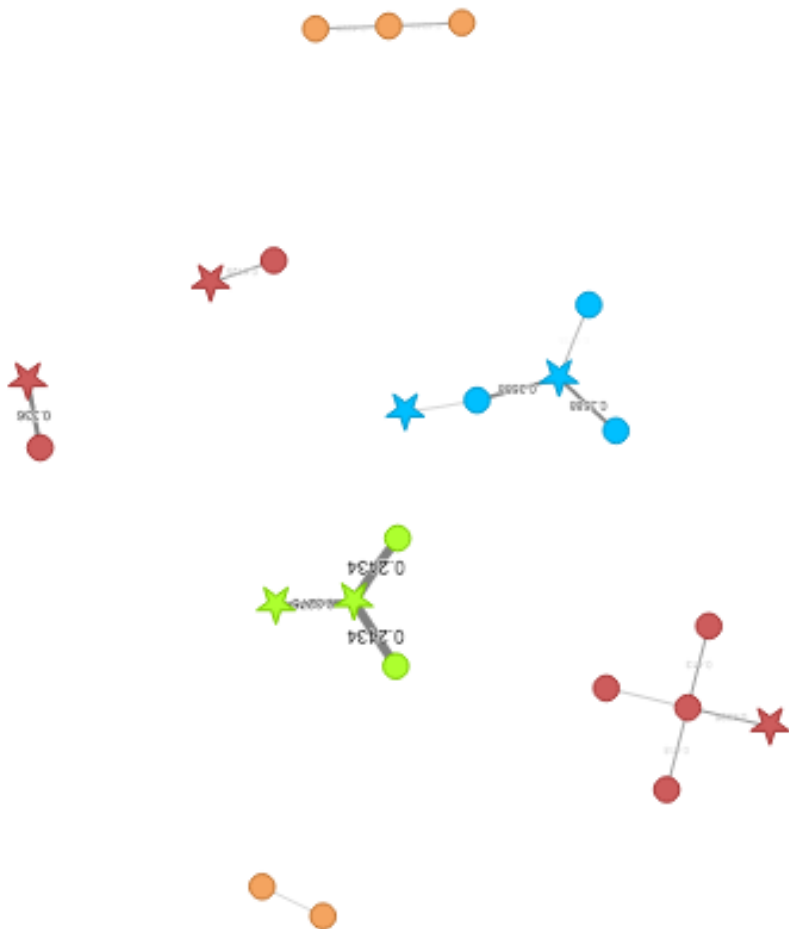
Gambar 8.131 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Perkapalan ITS dengan nilai K sama dengan



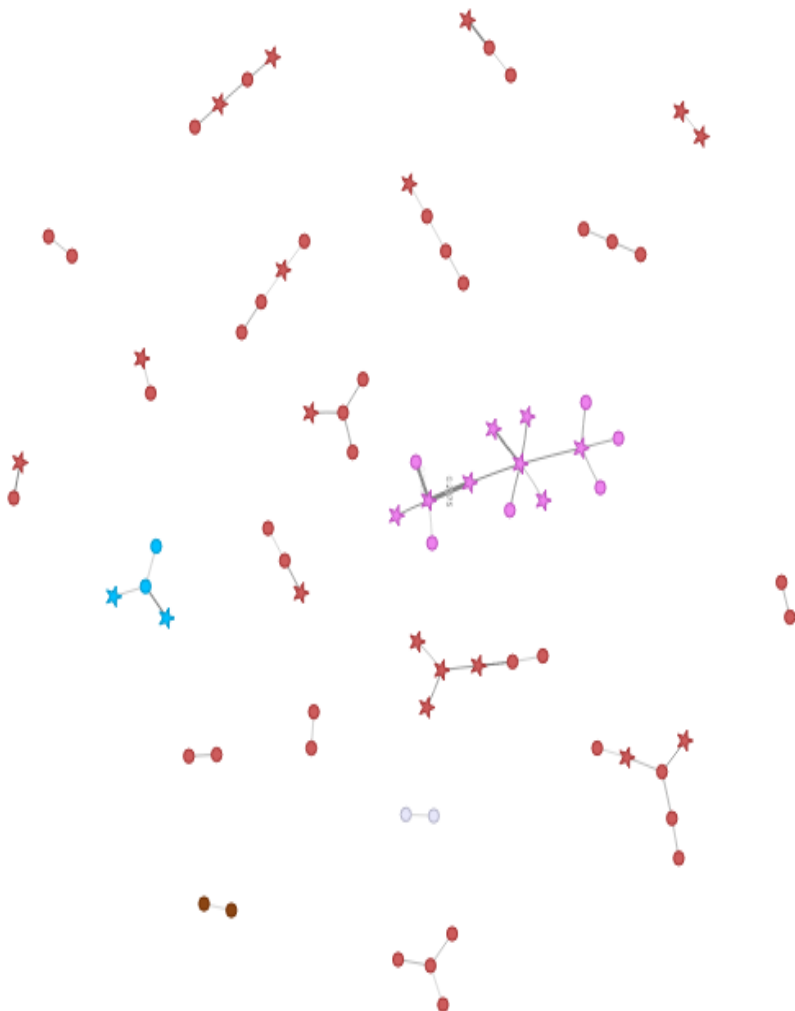
Gambar 8.132 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Sistem Perkapalan ITS dengan nilai K sama dengan 10



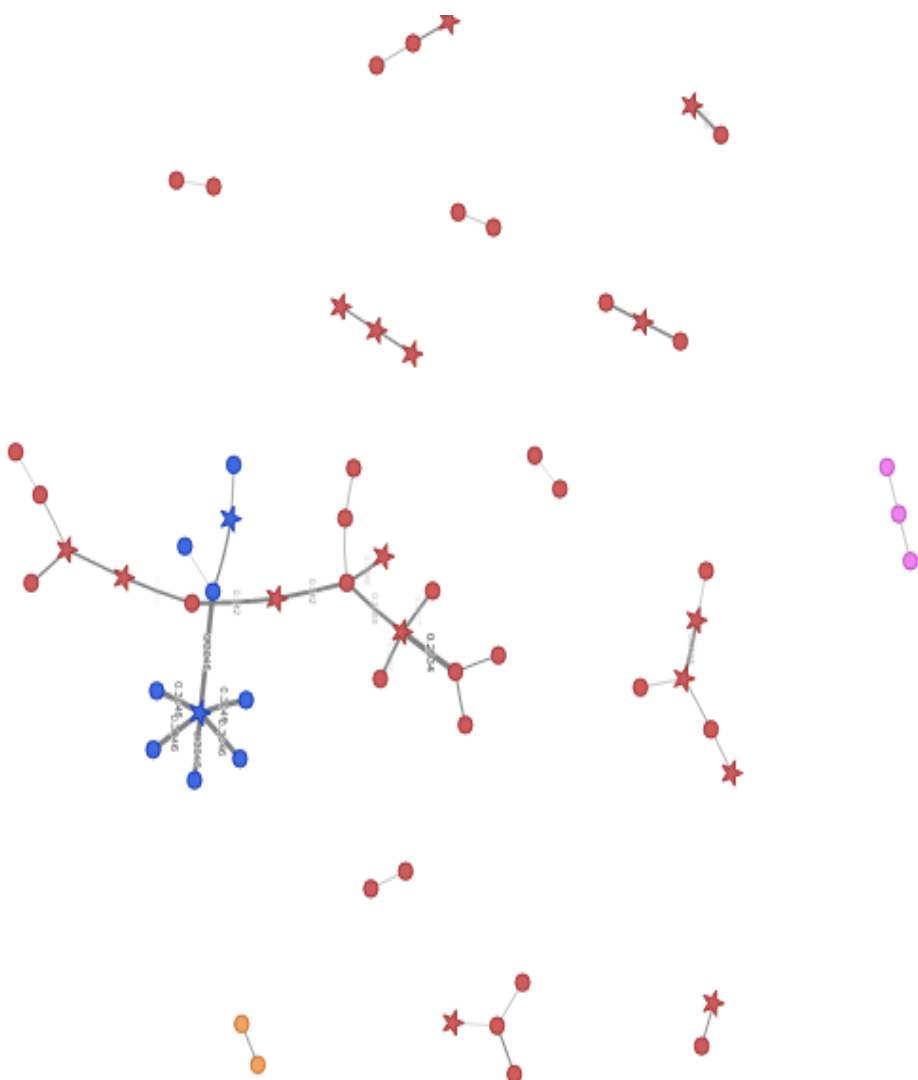
Gambar 8.133 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Kelautan ITS dengan nilai K sama dengan 10



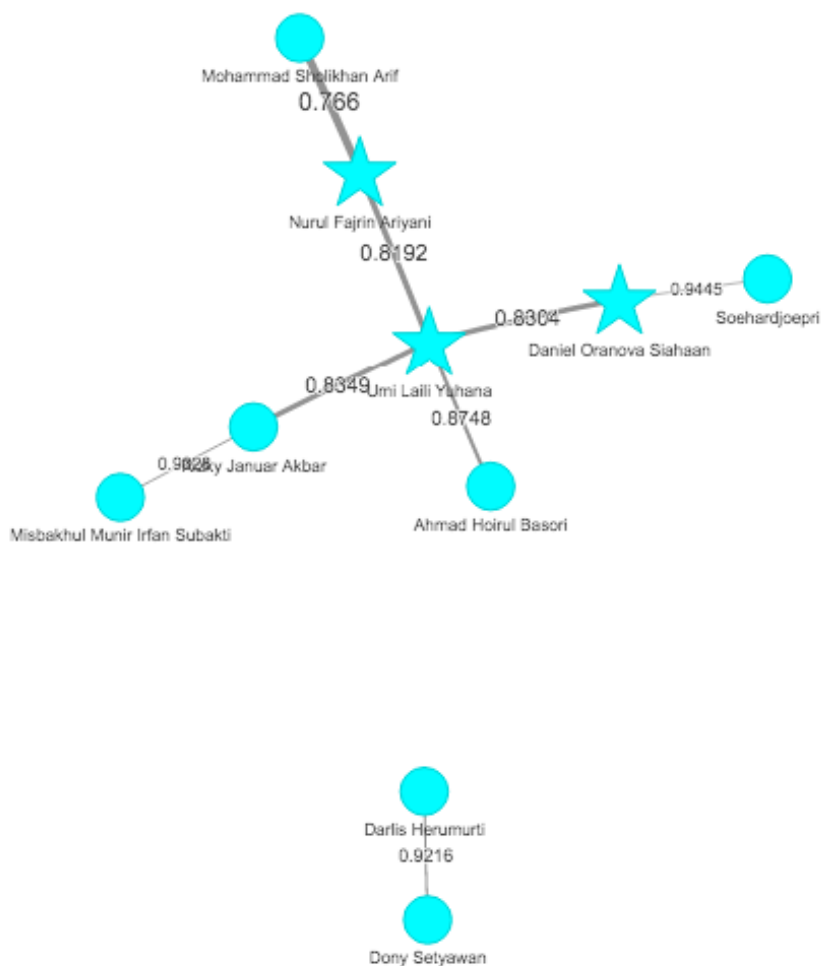
Gambar 8.134 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Transportasi Laut ITS dengan nilai K sama dengan 10



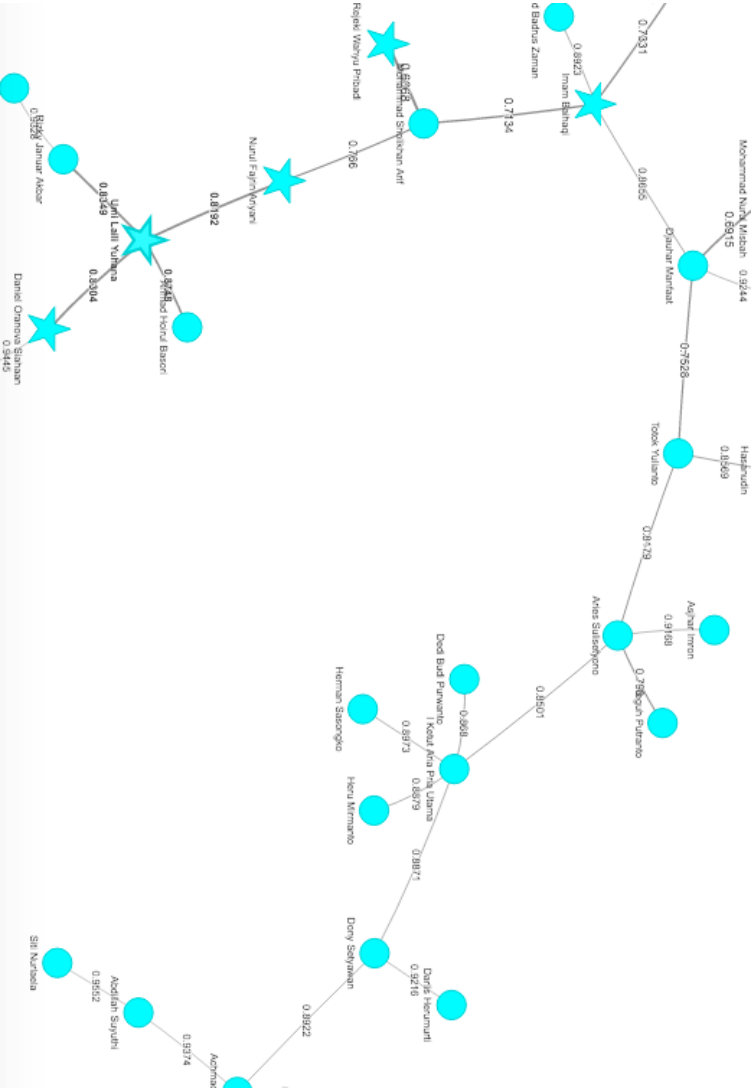
Gambar 8.135 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Teknik Informatika ITS dengan nilai K sama dengan



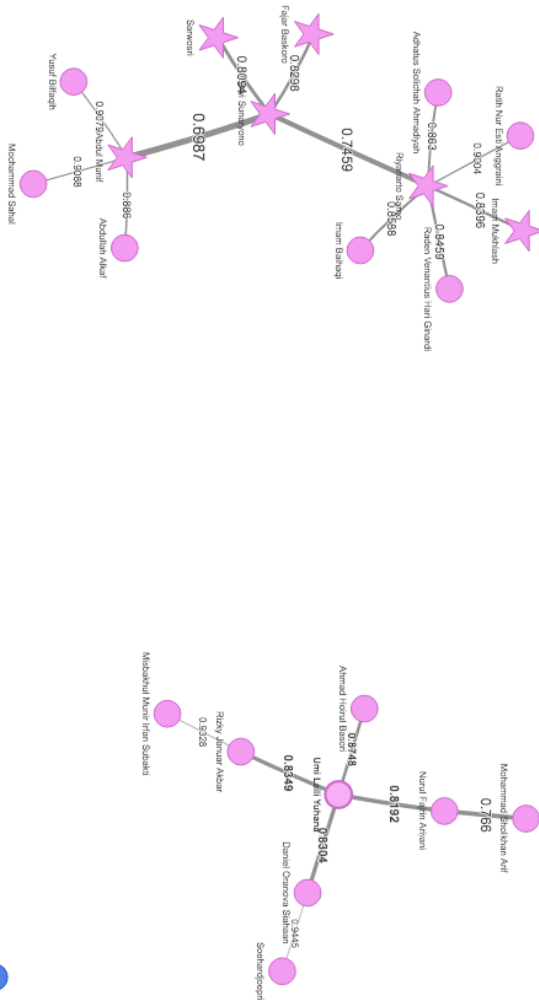
Gambar 8.136 Gambar graf cluster peneliti ekspansi sinonim kata jurusan Sistem Informasi ITS dengan nilai K sama dengan 10



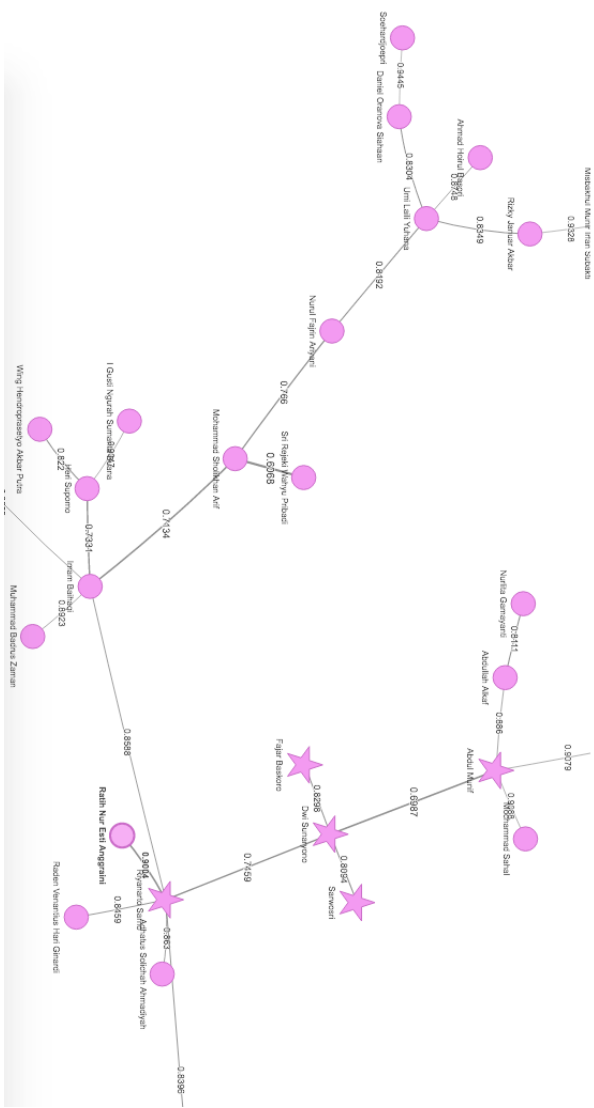
Gambar 8.137 Gambar cluster kerjasama yang terpecah dalam graf cluster kerjasama peneliti FTIF ITS



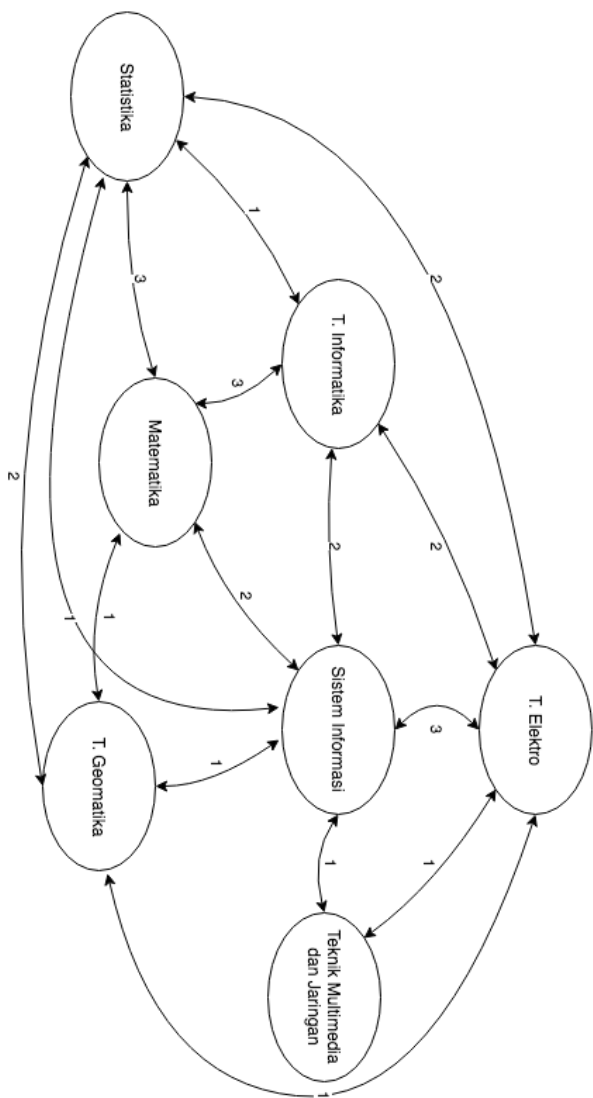
Gambar 8.138 Gambar cluster kerjasama keseluruhan dalam graf kerjasama peneliti ITS



Gambar 8.139 Gambar cluster kerjasama yang terpecah dalam graf cluster kerjasama peneliti FTIF ITS



Gambar 8.140 Gambar cluster kerjasama keseluruhan dalam graf kerjasama peneliti ITS



Gambar 8.141 Keterkaitan antar jurusan yang memiliki kemiripan fokus penelitian

8.2 Lampiran Tabel

Tabel 8.1 Tabel daftar 10 peneliti tertinggi yang memiliki term penelitian tidak termasuk dalam sinonim kata

No	Nama Dosen	Kode Jurusan	Term tidak termasuk dalam sinonim	Nilai Betweenness
1	Dimas Anton Asfani	22	166	0
2	Imam Arifin	22	138	0
3	Murni Rachmawati	32	132	0
4	Nanik Suciati	51	127	0
5	Soedibyo	22	122	6031
6	Siti Nurkhamidah	23	115	0
7	Indah Trisnawati Dwi Tjahjaningrum	15	114	0
8	Eko Nurmianto	25	113	0
9	Yudha Prasetyawan	25	111	0
10	Muhammad Taufik	35	111	0

Tabel 8.2 Tabel daftar 90 term tertinggi berdasarkan nilai IDF dari judul penelitian yang tidak terdapat pada kamus kata kateglo

No	Term	IDF
1	years	2.671173
2	blimbing	2.671173
3	ddos	2.671173
4	ekf-ui-wdf	2.671173
5	toksik	2.671173
6	activation	2.671173
7	type-2	2.671173

8	bo	2.671173
9	identities	2.671173
10	soca	2.671173
11	x-carbon	2.671173
12	ebu	2.671173
13	missile	2.671173
14	s-2	2.671173
15	steganography	2.671173
16	icici	2.671173
17	uniquac	2.671173
18	konsentris	2.671173
19	sphere	2.671173
20	implication	2.671173
21	equilibrium	2.671173
22	recast	2.671173
23	javascript	2.671173
24	webometrics	2.671173
25	fillers	2.671173
26	springer	2.671173
27	realtime	2.671173
28	flue	2.671173
29	finance	2.671173
30	suitability	2.671173
31	impulse	2.671173
32	mengoptimasi	2.671173
33	implant	2.671173
34	samarinda	2.671173
35	tambaksari	2.671173
36	piezoelectric	2.671173
37	internasional	2.671173

38	icaf	2.671173
39	aromatik	2.671173
40	nutrient	2.671173
41	multimodal	2.671173
42	millimeter-wave	2.671173
43	equipped	2.671173
44	glomerata	2.671173
45	perisa	2.671173
46	maintainability	2.671173
47	trees	2.671173
48	catamarans	2.671173
49	segmented	2.671173
50	validation	2.671173
51	sketchup	2.671173
52	grossus	2.671173
53	pool	2.671173
54	requirements	2.671173
55	stmik	2.671173
56	amobilisasi	2.671173
57	fun	2.671173
58	senvar	2.671173
59	seamless	2.671173
60	daylighting	2.671173
61	pan hcl fe3o4	2.671173
62	maneuverability	2.671173
63	computasional	2.671173
64	initiatives	2.671173
65	sub-and	2.671173
66	quarter	2.671173
67	dawt	2.671173

68	pex	2.671173
69	bumiputera	2.671173
70	wikibudaya	2.671173
71	webqual	2.671173
72	ystem	2.671173
73	trayektori	2.671173
74	multifunction	2.671173
75	hypogea	2.671173
76	selotapak	2.671173
77	firms	2.671173
78	categorical	2.671173
79	payangan-gianyar	2.671173
80	rul	2.671173
81	langka	2.671173
82	brondong	2.671173
83	pneumatic	2.671173
84	ligand	2.671173
85	autocad	2.671173
86	hydrocracking	2.671173
87	replacement	2.671173
88	transshipment	2.671173
89	sublasi	2.671173
90	kontras	2.671173

Tabel 8.3 Tabel term topik terbaik yang mewakili dosen FTIF

No	Nama Dosen	Term topik
1	Radityo Prasetyanto Wibowo	grow-able, healthcare, holobis, gamification, intelijensia, ohis, xml-rpc, suite, comparation, healthcare, visitors
2	Fajar Baskoro	non-textual, question answering, answer, predictor
3	Ahmad Muklason	pengmbangan, alpha
4	Febriliyan Samopa	kantuk, bendahara, tire, away3d, nopem, emban, syamrabu, jadid, rato, histograms, viewer, perobe, syarifah, php, kustomisasi, ambami, jenderal, interaktif, cuti, mysql, brain-computer, eksterior, ebu, menggunaka, cortical, mandibular, unreal, pt x, unity, bci
5	Anisah Herdiyanti Prabowo	25030, dharma, v3, itil, si ti, pmi, ieom, darah
6	Isye Ariesianti	ovarian, kernel, cancer, neighbors, edema, integrative, homology, macular, fundus, buluh, disulfide, worlds, feeds, twitter, surgery, k-medoids, fucntion, seldi-tof-ms, news, brings, one-pass, recognizing, kd-tree, protein, trending, ciri, bangkrut, completed, retina, melanoma, case-study, sequence
7	Faizal Mahananto	heart, variability, breathing, reflect, intensive high, respect, autonomic, posture, long-term, cardiac, care
8	Raden Venantius Hari Ginardi	urinary, yadika, leukocytes, praproses, arimoto, agglomerasi, togaf, berlian, cmmi-dev, dipstick, koteks, 50001 2011, urin,

		informasi teknologi, examination, ads-b, infection, multi-receiver, dr eam, tract, bangil, sim-d, nitrites, cobit, entropy, stmik, tomat, marker, correction, ti
9	Aris Tjahyanto	bpk-ri, features, gamelan, instrument, pesantren, connect, kobangdikal, kendari, signals, relief-f, sintetik, 9001 2000, fft-based, analysis-by-synthesis, spectral-based, machines, mmt-its, sound, instruments, note, time-frequency, tuner
10	Amalia Utamima	row, chromosome, facility, layout, solving, etikal, www sciencedirect com science article pii s0360835213001800, jot ntust edu tw index php jot article view 109, aco, ga, gallon, farmasi
11	Wahyu Suadi	silabus, ftif, ids, linux, honey, balancing, intrusion, useless, fidelit, phone, upta, best-effort, transmi, opnet, informatika, video, ip, ggunakan, mysql, packet, edition, round-robin, proxy, load
12	Yudhi Purwananto	algorit, c-means, hm, latar, segmentasi, const, partitions, delaunay, isye, purwananto, cubic, possibility, tchebichef, job-shop, sprague-grundy, kuantisasi, warna, rained, super-resolusi, rubic, nim, rcga, triangulation, arieshanti, nxt, yudhi, bangkrut, ragam, coded, itsbatik, integer
13	Nisfu Asrul Sani	logs, enelitian, event, unity, tridharma, mining-xml

14	Bekti Cahyo Hidayanto	3299 it2 7 pm 2011, erintegrasi, broadband, ace, divisi, lm, edutainment, communications
15	Edwin Riksakomara	standart, dipanegara, stimik
16	Adhatus Solichah Ahmadiyah	yu-chi, ei, solichah, anaglyph, 3dip, adhatus, francisco, lai, ahmadiyah, 24th, guan-ming, ist spie, kai-lung, hua, stereo
17	F.X. Arunanto	bounch, retas, paralelisasi, fp-growth, ngl, hadoop, mpich, 2011 1011, glpk, untukmenyelesaikan, lzss, mapreduce, cuda, nvidia, kaidah, branch, mutakhir, istmet
18	Imam Kuswardayan	sdk, skj, ekspres, smasher, kinect, reality, augmented, stikom, game, defense, crafter, microsof, wiimote, ngendali, openni, realm, backtraking, stemming, fid, bbs, ucial, por, libgdx, kit, maze, crosoft, skor, morpg, cms, vuforia, chomical, ant, android, unreal, prajurit, shooter, bulletin
19	Nur Aini Rakhmawati	linked, healthcare, visualization, islamic, oss, pindai, crowd-sourcing, monvis-musrenbang, mosque, mail, teens, selling, vocabulary, clam, informatka, beowulf, institutions, visualisation, issues, musrenbang, buying
20	Nanik Suciati	ke-3, polygon, ke-5, strip, representation, grouping, descriptor, busana, irregular, search, ke-2, it s, iciss, 978-979-96700-6-9, sub-band, co-occurence, serasi, icoict, neighbour, translation, dotted-board, clothing, generic, apcase, fcm, june, batik,

		packing, motif, extended, 28, texture, 13, aided, 1693-6930, fabrics, micro-structure, iceas, sign, ict, fringe, creation, ic3ina, cardinal, 2d, feature
21	Hanim Maria Astuti	success, scm, entitled, manajemen, systems, cba, 730-2002, hospital, erp, agribisnis, kud, incoming, ieom, classify, alpha
22	Wijayanti Nurul Khotimah	radiography, posterior, numbering, dental, mesiodistal, neck, 2 24, committee, information-based, sitting, post-stroke, tsukuba, fractures, detection link, desemer, ndonesian, kobe, jaci, standing, www fujipress jp finder xslt php, information-based, ibisml, scopus, terjemah, mode presentinputfile jaci001800040021 xml, radiographs, 15th, aquaculture, tokyo, patients, isyarat, panoramic
23	Rully Agus Hendrawan	standalone, mart, sisvo, utm, isico, tools, kp-2014-0076, Insentif, stnas, consuming, bi-dashboard, realisasi, analyze, independent, electricity, flexy, sub-, kppn
24	Mudjahidin	gears, os-commerce, google, oscommerce, implementasi, hibernate, unilever, jasper, rails, Implementation, afd, whyte, wcag, tasikmadu, colomadu, igniter, bytheway, sisspro, toko, cuti, 2 0, servqual, webometrics, flower, sitem, lkpp, ips, pg, webqual, kppn, e-katalog
25	Arif Djunaidy	cloud, signatures, encrypted, yz, koloni, hibridisasi, pengelol, off-line, mamd, kronis, abc-fuzzy, obstructif, tirun, ani, meaningful,

		janin, andard, gali, rfm, offline, apotek, otentikasi, konsultasi
26	Sarwosri	semantik, foundation, ponsel, nikasi, fragments, dailydrink, wsdl, komu, ekomendasi, trafficinfo, rss, supertrast, jsf, asisten, dankerangka, windows, ranking, dependency, udinus
27	Achmad Holil Noor Ali	abcd, bos, inpatient, ltll, departements, mendatan, sl tl, rl, senjang, ictil, scorecord, restructuring, afvoe, outpatients, fiskaria, rose, 2010-2015, balanced, bsc, portfolio, cmmi, bpk
28	Ratih Nur Esti Anggraini	ensiklopedia, pkk, recurrence, toddler, infant, nutrition, bayi, storing, kms-cs, manage, pregnancy, 14-16, jep, jetp, complexity, rpst, earthquake-tsunami, lampir, pkk-man, encyclopedia, e-pkk, hamil, posyandu, predict, recommendation, metrics, provision
29	Suhadi Lili	xna, multiplayer, milk, game, spac, playing, subsistem, lelang, shooter
30	Ahmad Saikhu	penjadualan, dokter, medis, darurat, dct, sha-1, 256, uh, cybernatika, stegan, ografi, sisip, satek, snastia, ubaya, enkripsi, bipartit, rijndael, bit, hig, unila, des, representasi, colony, integer, situasi
31	Muchammad Husni	mengindari, rijnda, selundup, dokeos, lacak, ukt, palettes, vpn, lts, srtp, kebocoranbbm, sql, ultrasonic, optocoupler, steganografi, mikrokontrol

32	Feby Artwodini Muqtadiroh	share-its, wikibudaya, wiki, hore, zak, non-curah, qlikview, virtualnusanantara, happy, pentaho, informatif, interactive, arif, requirements, grafik, pertanggungjawaban, tv9, metrics, refactoring, readysset, fusion, waris
33	Baskoro Adi Pratomo	voting, n-gram, tri-axial, gyroscope, falls, cosine, attacker s, accelerometer, intrusi, exposing, istmet
34	Eko Wahyu Tyas Darmaningrat	sop, permenkeu, fransisco, 169 pmk 01 2012, tugas kerja, 21-25, itil, mapinfo
35	Sholiq	effort, buton, logs, mining-xml, workflow, dinas, ti si, snasti, informaation, pengearuh, bloobis, event, portfolio, hortikultura, grafik, pertanggungjawaban, agribisnis, bu, readysset, investasi
36	Wiwik Anggraeni	deskriptif, habibah, thd, busana, sales, cision, taichung, autregression, bln, pert, akre, cis, 2087-0639, pasien, griya, nomer, ban, ftif, residence, kkp3n, multi-mode, dipanegara, stimik
37	Waskitho Wibisono	waskitho, wibisono, tus, publish-subscribe, programmi, iwang, botnets, dempster, kcic, phishing, multicast, backup, firefox, addons, networking, implosion, dns, mozilla, response-based, alimudin, browser, indonesian-japanese, jaringan, context-awareness, iceet, denial, naive-bayesian, knn, shafer, taut, event-based, konten, protocol, studiawan, creation, akhmad,

		reporting, tohari, hudan, puzzle, royyana, ddos
38	Mahendrawathi ER.	dashboard, vord, metatrader, spreadsheet, xzy, definition, fulfilment, viewpoints, pt z, aec, mining, customer, requirements, bpk
39	Chastine Fatichah	contrast, moments, hypergraph-partitioning, epileptic, l a b, newfm, aggregation, lossless, objek, boosying, co-occurrence, seizure, co-authorship, reversible, interest-based, analysis-based, vein, periapical, adaboost, radiograph, white, tebu, blood, ciri, sempurna, itsbatik, dental, daun, feature
40	Dwi Sunaryono	1978-0087, vaadin, scan, monta, ptn, snm, oktober-2013, geovisual, perfektif, song, lirik, dsms, asp net, ormatika, seelction, metro, rdbms, februari-2013, utama maka, berg, pppl, erak, tuntun, inf, ios, workflow, android, emotion, appointment, discover, recommendation, dongeng, cash, duplikasi, ledger, html, apllication, transaksi
41	Arif Wibisono	undergraduate, curriculum, programs, vol 17, 2012 2, jounal, human-resource, 53-74, year, dissemination, disruption, e-business, isico
42	Dini Adni Navastara	thresholding, fuzziness, teeth, multistage, kirim, computer-aided, trabecular, mandibular, panoramic, dental
43	Riyanarto Sarno	soad, account, oriented, payable, soa, customizable, asset, servis, enterprise, servi, tecture, receivabl, embangunan,

		receivable, koreografi, archi, hitecture, customizzable, orkestrasi, automation, perusahaa, resource, architecture, service, ce, manufakturing, ledger, reporting, similiarity, cash, non-free, crm, fixed
44	Daniel Oranova Siahaan	card, relevant, simialirity, conflict-based, rad, hybrids, syntatic-semantic, e-purse, textual, multi-capacity, customer-oriented, cots, indexing, freeway, kalimat, book, numbers, redundansi, benefits, fit, e-government
45	Ary Mazharuddin Shiddiqi	pesan, plug-in, k-192, moslem, rsa, mms, e-incubator, safer, low-rate, digilib, cpr, serang, tode, unggah, congestion, ddos, mikrokontrolle, html
46	Supeno Djanali	djanali, supeno, protokol, attacker, routing, raspberry, honeypot, shiddiqi, aodv ext, prolonging, tsepo, subsystem, ary, pa-short, dsr, qos, kongesti, surtana, henning, aware, expiration, diffserf, ciptaningtyas, oslr, intserv, refreshing, nkalai, aodv-rd, mesh, sql, wireless, umts, ims, scalable, studiawan, hudan, ip, honey, analyzing, behaviour, cluster
47	Handayani Tjandrasa	non-proliferative, retinopathy, diabetic, exudates, multi-temporal, inpainting, haar-like, hemorrhage, gradien, hypothesis, wavelets, kinematik, laplacian, averaged, pyramid, lesion, plantation, self-organizing, farming, ikonos, stages, margin, extracted, radially, segmented, wajah

48	Abdul Munif	disambiguation, sense, ibukreatif, fabrics, fringe
49	Anny Yuniarti	text, radiographs, individuals, radiography, posterior, arabic, histogram, panoramic, gray, wijayanti, pdt2, dictionary, equaliz, pdt, representatif, khotimah, ation, region-based, positional, numbering, decimation-free, age, dental, photometric, anny, geometric, vol 11, yudhi, multi-dokumen, rectification, kontras, linearly, yuniarti, zainal, distorted, curved, cardinal, fracture, mar, stereo, representation, tes, directional
50	Danu Pranantha	exploiting, electoral, biometrics, scalable
51	Tony Dwi Susanto	sms-based, e-government, egovernment, acceptance, services, les, individual, adoption, itil, 1203-3294, citizen, quarterly, icieis2013, universiti, acceptation, telescope, non-adopters, cobl, isico2013, billing, sms-mias, adopters, texte, publics, 1929-3348, differences, kab x, no 1-2, facteurs, elsevier, 486-497, sms-obm, prestation, v 3, v3, sesindo, pt xyz, government, 52, pmi, utm, si ti, vo, permenpan, kaliwates, factors, darah, prosedur
52	Dr. Apol Pribadi Subriadi	firm, usability, prioritizing, icbb, intencity, e-goverment, bapakism, projects, scales, www.jatit.org/volumes/vol59no3/24vol59no3.pdf, resource-based, activity-based, mpti, pihakpihak, paradox, matang,

		effort, contribution, continuity, webqual, presenter, website
53	Arya Yudhi Wijaya	pseudopolar, vol 18, jurnaltelkomnika, 649-657
54	Joko Lianto Buliali	lianto, rdf, buliali, srf02, joko, tl, transceiver, polinema, c4 5, mandala, karisma, makayasa, 2338-5154, upt, vision, silvester, victor, atmega8a, tena, saikhu, kadet, 20000 2005, santa, koefisen, bgr, iso lec, katolik, jani, muharom, graphical, trinanda, alstom, 27001 2005, bilqis, puskom, fredie, phi, lsq lec, syahri, agnes, acak, itil, hariadi, ntt, amalia, chi-square, adhi, gaji, bilang
55	Umi Laili Yuhana	academic, instrument, cari, kolaborasi, thresholds, georafis, 9126, context-aware, izin, peraturan, administration, ontology-based, rayonisasi, madya, googlemaps, portability, upload, pemanfaatan, evolvability, module, characteristic, metric, koleksi, proposed, metadata, iso iec, reklame, establishing
56	Amna Shifia Nisafani	elearning, on-the-fly, rds, culture-vid, dmr, performance-aware, no 52, undergraduate, permenpan, curriculum, wikibudaya
57	Andre Parvian Aristio	approval, 4 1, pulih, deputi, val sal, 27000, kemenko, polhukam, list subcon, 27002, koperasi, iso, adhi, cobit
58	Faizal Johan Atletiko	hatma, faizal, indoriko, shin, radityo, elekto, samopa, atletiko, febriliyan, attetion, suryotrisongko, setiawan,

		brainwave, electroencephalographic, analyzing
59	Diana Purwitasari	summarization, sentence, multi-document, storet, fiqh, sentences, ginardi, overlapping, relatedness, updated, mahaputra, coherence, cek, query-entropy, formulti-document, trqe, color-texture-based, admin, term, blood, topic, weighting, trending, bobot
60	Hatma Suryotrisongko	driving, semantik, drowsiness, sleep, deprivation, android-based, chronology, reactiontest, greenfoot, kemudi, isico, semantic, brainwave, qr, pditt, attention, electroencephalographic, brain-computer, clinical, bci, analyzing
61	Agus Zainal Arifin	zainal, sugiyanto, arab, genre, agus, sentence, multi-document, weighting, summarization, arifin, kalimat, kategorisasi, rozi, randy, erliyah, osteopo, jannah, okumen, subject, 227-234, nanang, naser, eranti, infomation, cahya, cosfire, iceic, fakhrur, wihandika, 27-34, jawas, decimation-free, term, bone, teks, ranking, 12th, anny, cortical, multi-dokumen, rectification, linearly, yuniarti, artmap, topic, distorted, curved, hibridasi, tesa, trabecular, bobot, directional, dental
62	Khakim Ghozali	e-katalog, khakim, ghozali, instansi, lkpp, tv9, paybri, pemograman, sarno, asrul, 001 2009, schoolmap, po, mudjahidin, nisfu, lowong, riyanto, dekaloros, chandra,

		telematika, pemkot, otonom, sani, client, portofolio, dinas, perintah, unified, telkomsel, ess, iso iec, similiarity, jasa
63	Henning Titi Ciptaningtyas	arduino, evoting, qr-code, election s, piksel, akuntansi, portabel, rejoy, bigbluebutton, bkm, coro, bidireksional, signalized, graph-based, fuse, recording, samba, moodle, umts, cctv, steganografi, duration, ngagel, seamless, mulyo, rijndael, securing, modulus
64	Misbakhul Munir Irfan Subakti	variable-centered, null, table, values, simulated, rule, chinese, oracles, revisions, hongkong, trigger, vcirs, cases, reconstructing, mutating, generating, annealing, estimating, rules, perspectives, dependency
65	Renny Pradina Kusumawardani	argumentative, ahp-topsis, manager, linguistik, classifiers, zoning, non-formal, rbf, onkologi, gallon, mapinfo, komputasional
66	Nurul Fajrin Ariyani	spatio-temporal, ontologi, ontology, artifacts, 020728 64 it2 11 pn 08 2015, linked, metadata
67	Tohari Ahmad	fraud, elliptic, templates, fingerprint, cancelable, secret, cryptography, hiding, process-based, adjusted, string-based, minutiae, de-based, protecting, invertible, smoothness, grayscale, pixels, x 509, certificate, expansion-based, neighbouring, increasing, curve, steganography, securing,

		rde-based, blocks, madm, revisited, security
68	Bilqis Amaliah	boarding, maskapai, 23, semantik, vogel s, voted, approximation, akakom, lipstick, integral, 15 16, sriti, perceptron, 3th, socdic, kanke, aided, stmik, udinus, stikom, melanoma
69	Darlis Herumurti	dsm, rgb, radon, resolution, road, osteoporosis, mandible, seeded, clues, shadow, countour, based-on, extraction, operations, art, aerial, morphological, artmap
70	Siti Rochimah	ibukreatif, facebook, rochimah, systematic, apps, konsistensi, multi-perspective, non-source, arwan, legacy, descriptions, reengineering, sekuensial, plugin, m-kms, standardized, bug, munif, service-based, emicu, architectures, vico, websites, kreativitas, juti, literature, diagram, ade, jejaring, refactoring, migration, academic, siti
71	Rizky Januar Akbar	categorization, completion, rasional, stackoverflow, repositories, usage, patterns, lda
72	Rully Sulaiman	nonlinier, fletcher, ellipsoid, rekuren, blurred, chaos, biometrical, goldfarb, gene, kutta-fungsi, shanno, steepest, fraktal, least-square, bounded, descent, broyden, entry, recurrent, branch, geometrical, companies, convex, geometric, pairs, recursive, inventori, microarray, runge

73	Irmasari Hafidz	telepon, pyxera, cendekia, pyxera-dow, america, united, som
74	Erma Suryani	beef, demand, fulfillment, uncertainties, cope, cow s, model-driven, cpo, cmc, fulfill, strategic, supply, cargo, soybean, crude
75	Victor Hariadi	citacee, nimum, suffix, voucher, menga, bioskop, aplikasi, workshop-conference, widget, spanning, ticketing, e-payment, diponegoro, capacitated, lam, smasher, transaksi
76	Bambang Setiawan	crm, attention, drowsiness, payment, serba, guideliness, accesbility, ortopedi, mechine, worker, mayur, agrobisnis, mediation, gateway, webometrics, notifikasi, quick, edition, ystem, gres, isico
77	Hudan Studiawan	honeypot, attacker, raspberry, 013537 it2 7 pn 08 01 2013, email-based, trapping, attackers, fingerprint-based, cluster, steganography, cryptography, exposing
78	Raras Tyasnurita	hankuk
79	Radityo Anggoro	on-demand, environments, aodv, hoc, vanet, geopps, multipath, vadd, storm, aomdv, broadcast, non-dtn, dtn, manet, gsr, vanets, ad, distance, relay, probabilistic, protokol, adobe, non- proliferative, fault-tolerant, protocols, segmented
80	Retno Aulia Vinarti	heart, disease, cassava, visualization, isico, erp, pd-dash, breast, gali, security,

		stepwise, pt xyz, comparation, visitors, ganas, food, early, tree
81	Royyana Muslim Ijtihadie	wibisono, muslim, vol 114, cybernetics, holil, i r, mailing-list, cyberneticscom, 145, usagawa, no 53, tsuyoshi, chisaki, 141, 10 1109 cyberneticscom 2013 6865798, ieice, yoshifumi, et2014-6, innah, ijthadie, herbert, pp 27-30, 2014 5, waskitho, quad, overlay, tohari, affandi, synchronization, hiding, rde-based, royyana, ahmad

BIODATA PENULIS



Hifnie Bilfash, Lahir pada tanggal 13 September 1994. Anak pertama dari dua bersaudara. Saat ini sedang menempuh pendidikan perguruan tinggi negeri di Institut Teknologi Sepuluh Nopember Surabaya di jurusan Teknik Informatika, Fakultas Teknologi Informasi, angkatan tahun 2013.

Pernah mengikuti beberapa organisasi dan beberapa kepanitian diantaranya adalah Staf Pengembangan Profesi Himpunan Mahasiswa Teknik

Computer-Informatika 2014-2015 dan Kepala Departemen Pengembangan Profesi Himpunan Mahasiswa Teknik Computer-Informatika 2015-2016.

bilfash.official@gmail.com